



Jorge Salvador Paredes Merino

**Síntese Automática de Sistemas de Inferência
Fuzzy para Classificação**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Processamento de Sinais e Controle do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio.

Orientador : Prof. Ricardo Tanscheit
Coorientador: Dr. Marley Maria Bernardes Rebuzzi Vellasco

Rio de Janeiro
Dezembro de 2015



Jorge Salvador Paredes Merino

**Síntese Automática de Sistemas de Inferência
Fuzzy para Classificação**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Processamento de Sinais e Controle do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Ricardo Tanscheit

Orientador

Departamento de Engenharia Elétrica – PUC-Rio

Dr. Marley Maria Bernardes Rebuszi Vellasco

Coorientador

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Jorge Luís Machado do Amaral

Faculdade de Engenharia – UERJ

Prof. Karla Tereza Figueiredo Leite

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Márcio da Silveira Carvalho

Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 21 de Dezembro de 2015

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Jorge Salvador Paredes Merino

Graduou-se em Engenharia Mecatrónica pela Universidad Nacional de Ingeniería (Lima, Perú). Durante a graduação trabalhou principalmente em projetos relacionados a Visão Computacional aplicada à robótica móvel e trabalhos de bases de dados. Trabalhou na empresa ABB Perú S.A na área de Discrete Automation and Motion.

Ficha Catalográfica

Paredes Merino, Jorge Salvador

Síntese Automática de Sistemas de Inferência Fuzzy para Classificação / Jorge Salvador Paredes Merino; orientador: Ricardo Tanscheit; co-orientador: Marley Maria Bernardes Rebutzi Vellasco. – 2015.

106 f. : il. color.; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia.

1. Engenharia Elétrica – Tese. 2. Sistema de Inferência Fuzzy;. 3. Classificação de padrões;. 4. Extração automática de regras fuzzy.. I. Tanscheit, Ricardo. II. Vellasco, Marley Maria Bernardes Rebutzi. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

Agradecimentos

Aos meus orientadores Professores Marley Vellasco e Ricardo Tanscheit pelo apoio, simpatia de sempre, e incentivo para a realização deste trabalho.

Ao CNPq, CAPES e à PUC-Rio, pelos auxílios concedidos e estrutura concedida, sem os quais este trabalho não poderia ter sido realizado.

Aos meus pais, Jorge e Altamira, pela formação baseada nos valores, disciplina e amor. Além da compreensão e paciência nos momentos difíceis. Ao meu irmão David por ser um bom ouvinte e pelos ânimos.

Ao pessoal do DEE-PUC-Rio pela ajuda de todos os dias. Ao Laboratório de Visão Computacional, coordenado pelo Prof. Raul Feitosa, e em especial ao meu amigo Pedro Acchancaray pelo aporte dos recursos computacionais nos experimentos iniciais, o que permitiu avaliar restrições do modelo proposto.

Finalmente, ao meu amigo Adriano Koshiyama pelas recomendações, motivação e longas discussões acerca da presente dissertação.

Resumo

Paredes Merino, Jorge Salvador; Tanscheit, Ricardo; Vellasco, Marley Maria Bernardes Rebutzi. **Síntese Automática de Sistemas de Inferência Fuzzy para Classificação**. Rio de Janeiro, 2015. 106p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Hoje em dia, grande parte do conhecimento acumulado está armazenado em forma de dados. Para muitos problemas de classificação, tenta-se aprender a relação entre um conjunto de variáveis (atributos) e uma variável alvo de interesse. Dentre as ferramentas capazes de atuar como modelos representativos de sistemas reais, os Sistemas de Inferência Fuzzy são considerados excelentes com respeito à representação do conhecimento de forma compreensível, por serem baseados em regras linguísticas. Este quesito de interpretabilidade linguística é relevante em várias aplicações em que não se deseja apenas um modelo do tipo “caixa preta”, que, por mais precisão que proporcione, não fornece uma explicação de como os resultados são obtidos. Esta dissertação aborda o desenvolvimento de um Sistema de Inferência Fuzzy de forma automática, buscando uma base de regras que valorize a interpretabilidade linguística e que, ao mesmo tempo, forneça uma boa acurácia. Para tanto, é proposto o modelo AutoFIS-Class, um método automático para a geração de Sistemas de Inferência Fuzzy para problemas de classificação. As características do modelo são: (i) geração de premissas que garantam critérios mínimos de qualidade, (ii) associação de cada premissa a um termo conseqüente mais compatível e (iii) agregação de regras de uma mesma classe por meio de operadores que ponderem a influência de cada regra. O modelo proposto é avaliado em 45 bases de dados benchmark e seus resultados são comparados com modelos da literatura baseados em Algoritmos Evolucionários. Os resultados comprovam que o Sistema de Inferência gerado é competitivo, apresentando uma boa acurácia com um baixo número de regras.

Palavras-chave

Sistema de Inferência Fuzzy; Classificação de padrões; Extração automática de regras fuzzy.

Abstract

Paredes Merino, Jorge Salvador; Tanscheit, Ricardo (Advisor); Vellasco, Marley Maria Bernardes Rebuzzi. **Automatic Synthesis of Fuzzy Inference Systems for Classification**. Rio de Janeiro, 2015. 106p. MSc. Dissertation – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Nowadays, much of the accumulated knowledge is stored as data. In many classification problems the relationship between a set of variables (attributes) and a target variable of interest must be learned. Among the tools capable of modeling real systems, Fuzzy Inference Systems are considered excellent with respect to the knowledge representation in a comprehensible way, as they are based on inference rules. This is relevant in applications where a “black box” model does not suffice. This model may attain good accuracy, but does not explain how results are obtained. This dissertation presents the development of a Fuzzy Inference System in an automatic manner, where the rule base should favour linguistic interpretability and at the same time provide good accuracy. In this sense, this work proposes the AutoFIS-Class model, an automatic method for generating Fuzzy Inference Systems for classification problems. Its main features are: (i) generation of premises to ensure minimum, quality criteria, (ii) association of each rule premise to the most compatible consequent term; and (iii) aggregation of rules for each class through operator that weigh the relevance of each rule. The proposed model was evaluated for 45 datasets and their results were compared to existing models based on Evolutionary Algorithms. Results show that the proposed Fuzzy Inference System is competitive, presenting good accuracy with a low number of rules.

Keywords

Fuzzy Inference Systems; Pattern classification; Fuzzy rules extraction.

Sumário

1	Introdução	14
1.1	Objetivos	15
1.1.1	Objetivo Primário	15
1.1.2	Objetivos Secundários	15
1.2	Contribuições	15
1.3	Descrição e Organização da Dissertação	16
2	Sistemas de Inferência Fuzzy para Classificação	17
2.1	Sistemas Fuzzy Evolucionários	17
2.1.1	Arquitetura básica	18
2.1.2	Áreas de atuação	19
2.1.3	Abordagens para Base de Regras	20
2.1.4	Aplicações em Classificação	21
2.2	Sistemas Fuzzy Evolutivos	23
2.2.1	Sistema Evolutivo com Clusterização	24
2.2.2	Sistema Evolutivo com Árvore de Regressão Linear Fuzzy	26
2.3	Interpretabilidade	28
3	Modelo Automatic Fuzzy Inference System (AutoFIS)	31
3.1	Definições e Notações	32
3.2	Arquitetura	33
3.2.1	Fuzzificação	34
3.2.2	Formulação	36
3.2.3	Associação	40
3.2.3.1	Grau de Confiança	40
3.2.3.2	Grau de Confiança Penalizado	41
3.2.3.3	Frequência Máxima	41
3.2.3.4	Mínimos Quadrados Restritos	42
3.2.4	Agregação	43
3.2.4.1	Máximo	43
3.2.4.2	MQR	44
3.2.4.3	IntMQR	44
3.2.5	Decisão	44
3.3	Resumo	45
4	Estudo de Casos	46
4.1	Investigação Empírica da Arquitetura do Modelo AutoFIS-Class	49
4.1.1	Experimento 1 - Avaliação dos Operadores Fuzzy	51
4.1.2	Experimento 2 - Operadores de Agregação	56
4.1.3	Experimento 3 - Limiar do Filtro do Suporte	59
4.1.4	Experimento 4 - Métodos de Associação	62
4.1.5	Experimento 5 - Tipos de Funções de Pertinência e Filtro de Suporte	65
4.1.5.1	Bases Binárias	65
4.1.5.2	Múltiplas classes	68

4.1.6	Experimento 6 - Número de termos linguísticos	70
4.1.6.1	Bases Binárias	71
4.1.6.2	Múltiplas classes	73
4.1.7	Experimento 7 - Seleção de Filtros	74
4.1.7.1	Bases Binárias	75
4.1.7.2	Múltiplas classes	77
4.1.8	Experimento 8 - Tamanho da premissa	79
4.1.8.1	Bases Binárias	79
4.1.8.2	Múltiplas classes	83
4.1.9	Experimento 9 - Limiar do Filtro de Similaridade	85
4.1.9.1	Bases Binárias	85
4.1.9.2	Múltiplas classes	88
4.1.10	Resumo - Parâmetros das configuração obtidas	90
4.2	Extensão do modelo AutoFIS-Class para bases de dados de dimensão maior em atributos	91
4.2.1	Bases Binárias	91
4.2.2	Múltiplas classes	92
4.3	Comparações com outros modelos	94
4.3.1	Resultados e discussões	94
4.3.1.1	Grupo 1	94
4.3.1.2	Grupo 2	97
5	Conclusões e Trabalhos Futuros	100
	Referências Bibliográficas	102

Lista de figuras

2.1	Diagrama genérico de um SFE	18
2.2	Atuação da MHE em um SFE, adaptada de [18].	19
2.3	Abordagens de Bases de Regras de acordo a codificação do Indivíduo.	20
2.4	Conjuntos Fuzzy projetados pelos clusters, adaptada de [38]	25
2.5	Ilustração do potencial das amostras de dados, adaptada de [38]	26
2.6	Exemplo de árvore de regressão linear, adaptada de [38]	26
2.7	(a) Partição do espaço de entrada de uma árvore de regressão linear. (b) Partição do espaço de uma árvore de regressão linear fuzzy, adaptada de [38]	27
2.8	Indicadores nos Quadrantes de Interpretabilidade	29
3.1	Etapa de Inferência no modelo AutoFIS-Class.	34
3.2	Tipo de funções de pertinência triangular: (a) Fortemente Particionada e (b) Tukey.	35
3.3	Premissas agrupadas pelo tamanho das premissas para 3 atributos com 2 conjuntos fuzzy cada um.	37
3.4	Premissa que não excede o limiar do Suporte	38
3.5	Premissas dispensadas do espaço de busca	38
3.6	Comparação de CD com o PCD	41
3.7	Agregação usando o máximo.	43
3.8	Inter-relação das premissas sobreviventes da Associação	44
4.1	Delineamento sequencial dos experimentos com o modelo AutoFIS-Class	49
4.2	Experimento de avaliação da t-norma e operador de negação	51
4.3	Experimento de avaliação dos Operadores de Agregação.	56
4.4	Experimento de avaliação do limiar do Filtro de Suporte.	59
4.5	Experimento de avaliação dos Métodos de Associação	62
4.6	Experimento de funções de pertinência e tipo de suporte	65
4.7	Experimento de número de termos linguísticos.	70
4.8	Experimento com uso dos Filtros PCD e Similaridade.	75
4.9	Experimento com o tamanho da premissa.	79
4.10	Experimento do Limiar do Filtro de Similaridade.	85

Lista de tabelas

2.1	Indicadores dos Quadrantes de Interpretabilidade	30
3.1	Representação de uma base de dados	32
3.2	Associação de funções de pertinência a um atributo	32
3.3	Base de dados de Classificação ($k \in \{1, 2, 3\}$)	35
3.4	Fuzzificação dos atributos (não categóricos) e Binarização dos atributos categóricos e saída.	36
3.5	Premissas para analisar o efeito da Frequência Relativa	39
3.6	Efeito da diferença entre o Suporte Relativo e a Frequência Relativa	39
4.1	Principais características das bases de dados binárias.	47
4.2	Principais características das bases de dados de múltiplas classes.	48
4.3	Bases de dados utilizadas para gerar a arquitetura empírica do Modelo AutoFIS-Class.	50
4.4	Acurácias e número de regras no Experimento 1.	52
4.5	Resultados do teste de Friedman e Holm para a acurácia do Experimento 1.	53
4.6	Resultados do teste de Friedman e Holm para o número de regras do Experimento 1.	53
4.7	Fator de incremento em tempo por uso do operador de Negação nas configurações de Mínimo e Produto.	55
4.8	Acurácias das bases de dados no Experimento 1.	57
4.9	Teste de Friedman e Holm para a acurácia.	58
4.10	Teste de Friedman e Holm para o número de regras.	58
4.11	Acurácia e número de regras no Experimento 3.	60
4.12	Resultados do teste de Friedman e Holm para a acurácia do Experimento 3.	61
4.13	Resultados do teste de Friedman e Holm para o número de regras do Experimento 3.	61
4.14	Acurácias e número de regras no Experimento 4.	63
4.15	Acurácia e número de regras do Experimento 5 nas bases binárias.	66
4.16	Resultados do teste de Friedman e Holm para acurácias no Experimento 5 nas bases binárias.	66
4.17	Resultados do teste de Friedman para o número de regras do Experimento 5 nas bases binárias.	67
4.18	Acurácia e número de regras do Experimento 5 nas múltiplas classes.	68
4.19	Resultados do teste de Friedman e Holm para acurácias do Experimento 5 nas bases de múltiplas classes.	69
4.20	Resultados do teste de Friedman para o número de regras do Experimento 5 nas bases de múltiplas classes.	69
4.21	Acurácias e número de regras do Experimento 6 nas bases binárias.	71
4.22	Resultados do teste de Friedman e Holm para acurácias nas bases binárias do Experimento 6.	71
4.23	Resultados do teste de Friedman para o número de regras do Experimento 6 nas bases binárias.	72

4.24	Acurácias e número de regras do Experimento 6 nas bases de múltiplas classes.	73
4.25	Resultados do teste de Friedman e Holm para acurácias nas bases de múltiplas classes do Experimento 6.	73
4.26	Resultados do teste de Friedman para o número de regras nas bases de múltiplas classes do Experimento 6.	74
4.27	Acurácia e número de regras para o Experimento 7 nas bases binárias.	75
4.28	Resultados do teste de Friedman e Holm para acurácia nas bases binárias do Experimento 7.	76
4.29	Resultados do teste de Friedman para o número de regras do Experimento 7 nas bases binárias.	76
4.30	Acurácia e número de regras do Experimento 7 nas bases de múltiplas classes.	77
4.31	Resultados do teste de Friedman e Holm para acurácias nas bases de múltiplas classes do Experimento 7.	78
4.32	Resultados do teste de Friedman para o número de regras nas bases de múltiplas classes do Experimento 7.	78
4.33	Acurácias das bases de dados no Experimento 8 nas bases binárias.	80
4.34	Resultados do teste de Friedman e Holm para acurácias Experimento 8 nas bases binárias.	80
4.35	Resultados do teste de Friedman para o número de regras do Experimento 8 nas bases binárias.	81
4.36	Tempos empregados (segundos) pelas configurações no Experimento 8 nas bases binárias.	82
4.37	Acurácias e número de regras das bases de múltiplas classes no Experimento 8.	83
4.38	Resultados do teste de Friedman e Holm para acurácias nas bases de múltiplas classes do Experimento 8.	83
4.39	Resultados do teste de Friedman para o número de regras nas bases de múltiplas classes do Experimento 8.	84
4.40	Tempos empregados (segundos) pelas configurações no Experimento 8 nas bases de múltiplas classes.	84
4.41	Acurácias e número de regras para as bases binárias no Experimento 9.	86
4.42	Resultados do teste de Friedman e Holm para acurácias do Experimento 9 nas bases binárias.	86
4.43	Resultados do teste de Friedman para o número de regras do Experimento 9 nas bases binárias.	87
4.44	Acurácias e número de regras das bases de múltiplas classes no Experimento 9.	88
4.45	Resultados do teste de Friedman e Holm para acurácias Experimento 9 nas bases de múltiplas classes.	88
4.46	Resultados do teste de Friedman regras para o número de regras do Experimento 9 nas bases de múltiplas classes.	89
4.47	Configurações empíricas do modelo AutoFIS-Class.	90
4.48	Taxa de sucesso dos classificadores menores nas bases binárias.	91
4.49	Extensão do modelo AutoFIS em bases binárias.	92

4.50	Taxa de sucesso dos classificadores menores nas bases de múltiplas classes.	93
4.51	Extensão do modelo AutoFIS em bases de múltiplas classes.	93
4.52	Comparação de acurácias e número de regras no Grupo 1.	95
4.53	Resultados do teste de Friedman e Holm para acurácias no Grupo 1.	96
4.54	Resultados do teste de Friedman e Holm para o número de regras no Grupo 1.	96
4.55	Comparação de acurácias e número de regras no Grupo 2.	98
4.56	Resultados do teste de Friedman e Holm para a acurácia no Grupo 2.	99
4.57	Resultados do teste de Friedman e Holm para o número de regras no Grupo 2.	99

Cada livro, cada volume que vês, tem alma. A alma de quem o escreveu e a alma dos que o leram e viveram e sonharam com ele. Cada vez que um livro muda de mãos, cada vez que alguém desliza o olhar pelas suas páginas, o seu espírito cresce e torna-se forte.

Carlos Ruiz Zafón, *A sombra do vento*.

1

Introdução

Ao longo dos anos, muitos algoritmos foram desenvolvidos no campo de Aprendizado de Máquinas para facilitar a realização de diversas tarefas da vida cotidiana. Uma dessas tarefas é a classificação de padrões, com aplicações em diversas áreas, tais como Economia, Medicina, Visão Computacional, entre outras. Exemplos de algoritmos podem ser encontrados em [1, 2].

Dentre as ferramentas capazes de atuar como modelos representativos de sistemas reais, os Sistemas de Inferência Fuzzy são considerados excelentes com respeito à representação do conhecimento de forma humanamente compreensível, por meio de suas regras de inferência [3]. Esta questão é relevante em várias aplicações, pois não se deseja apenas um modelo do tipo “caixa preta”¹ [6], que, por mais precisão que proporcione, não fornece uma explicação de como os resultados são obtidos. Em outras palavras, além de acurácia, busca-se também interpretabilidade, cuja medição não é ainda totalmente estabelecida. Em Sistemas de Inferência Fuzzy, tais fatores dependem: da estrutura do modelo (Mamdani, Takagi-Sugeno-Kang, etc.), do número de regras, do número de elementos antecedentes e do formato das funções de pertinência, entre outros [7, 8].

Com o objetivo de construir sistemas que aliem acurácia à interpretabilidade, a maior parte dos trabalhos tem seguido dois caminhos: **(i)** uso de algoritmos evolucionários para a elaboração de bases de regras fuzzy, com ou sem ajuste das funções de pertinência [9–11], ou **(ii)** uso de Sistemas de Inferência Fuzzy Evolutivos [6, 12, 13], que criam e adaptam a base de regras fuzzy a partir do reagrupamento de novas observações. O ponto negativo da abordagem **(i)** é o seu alto custo computacional e o grande número de parâmetros adicionados em decorrência do uso de um algoritmo evolucionário – em alguns casos multi-objetivo. Já no caso da abordagem **(ii)**, ao não se presumir uma alocação inicial e fixa para as funções de pertinência, as regras geradas podem apresentar baixa interpretáveis.

Este trabalho combina o melhor de duas abordagens, isto é, a geração de um Sistema de Inferência Fuzzy de forma automática, buscando uma boa

¹Embora tenha havido esforços para desenvolver metodologias para abrir os modelos tipo “caixa preta” [4, 5], extraindo o conhecimento como regras, existem certas limitações e o processo não sempre é trivial.

acurácia, mas privilegiando a geração de uma base de regras que valorize a interpretação linguística. Propõe-se, portanto, o modelo AutoFIS-Class, um método determinístico e automático para a geração de Sistemas de Inferência Fuzzy com foco em Classificação. O AutoFIS-Class tem como principais características: (i) gerar premissas que garantam critérios mínimos de qualidade; (ii) associar cada premissa a um termo conseqüente mais compatível; e (iii) agregar regras para uma mesma classe por meio de operadores que ponderem cada regra com base na sua influência na classificação.

1.1 Objetivos

1.1.1 Objetivo Primário

O principal objetivo deste trabalho é propor um novo sintetizador automático genérico de Sistemas de Inferência Fuzzy para Classificação denominado **AutoFIS-Class**. Este sistema deve ser flexível, contemplando diferentes configurações, robusto, preciso e de fácil compreensão para o usuário.

1.1.2 Objetivos Secundários

1. Analisar as características do modelo AutoFIS em relação ao conjunto de Sistemas de Inferência Fuzzy existentes (terminologias, espectro, etc.).
2. Comparar os resultados do modelo AutoFIS com os de outros Sistemas Fuzzy Evolucionários.

1.2 Contribuições

A principal contribuição desta dissertação é a proposição e o desenvolvimento do novo modelo AutoFIS, com as seguintes características:

- Uso de uma abordagem do tipo determinística para a geração automática de premissas e regras nos Sistemas de Inferência Fuzzy (SIF);
- Uso de um operador pouco empregado: a Negação;
- Utilização de operadores de agregação que não apenas o Máximo para realizar agregar regras com um mesmo conseqüente;
- Elaboração de métricas de associação entre termos antecedentes e conseqüentes das regras;
- Estruturação do modelo para aplicação em outros campos, como, por exemplo, Regressão.

1.3 Descrição e Organização da Dissertação

Esta dissertação está assim organizada:

- O capítulo 2 realiza uma Revisão Bibliográfica, concentrando-se em Sistemas Fuzzy Evolucionários (SFE) e Sistemas Fuzzy Evolutivos. Nos SFE, são mencionadas heurísticas tais como os Algoritmos Genéticos (AG), Programação Genética (PG) e Multi-objetivos. Além disso, o capítulo trata do quesito Interpretabilidade nos SIF.
- O capítulo 3 apresenta o modelo AutoFIS-Class, discorrendo sobre os principais passos da síntese automática de um SIF: Fuzzificação, Formulação, Associação, Agregação, Decisão e Avaliação.
- O capítulo 4 descreve os experimentos realizados, assim como os resultados obtidos. O procedimento adotado consiste nas seguintes etapas: (i) avaliação das diferentes parametrizações do modelo AutoFIS-Class para um conjunto de benchmarks, de modo a avaliar o desempenho de cada um; (ii) a partir da melhor configuração geral, efetua-se a comparação com os demais SFEs para assim demonstrar que em aplicações tipo benchmark o modelo é tão eficiente quanto os demais SFEs presentes na literatura; (iii) elaboração de uma modelagem para um problema básico, ou seja, fornecer uma configuração por *default* ao usuário, visando à apresentar amplamente os resultados gerados pelo modelo, levando em conta a acurácia e interpretabilidade.
- Por fim, o capítulo 5 conclui o trabalho e discute os trabalhos futuros, sugerindo alguns novos direcionamentos, tanto para o aprimoramento do modelo, quanto em termos de potenciais aplicações e comparações inexploradas no trabalho.

Este capítulo apresenta os conceitos básicos de Sistemas de Inferência Fuzzy para Classificação. De acordo com [13], estes sistemas podem ser classificados em dois tipos:

- **Evolucionário:** compreende algoritmos que imitam os paradigmas da evolução, tais como mutação, crossover de cromossomos, reprodução e seleção.

Exemplos de metaheurísticas:

- Algoritmos Genéticos
 - Programação Genética
 - Algoritmos Evolucionários Multiobjetivos
 - Métodos de nichos
- **Evolutivo:** diz respeito ao desenvolvimento gradual do sistema, fazendo uso, por exemplo, de técnicas de agrupamento ou de árvores de regressão linear fuzzy.

O algoritmo proposto nesta dissertação é do tipo Evolutivo. Além disso, são apresentados de forma resumida os Sistemas Fuzzy Evolucionários, pois se usam algumas métricas, estratégias e conceitos comuns aos dois tipos no algoritmo proposto.

2.1 Sistemas Fuzzy Evolucionários

Os Sistemas de Inferência Fuzzy foram inicialmente concebidos para serem baseados no conhecimento de especialistas, mas, na década de 90, as abordagens orientadas a dados se tornaram mais presentes [13]. Uma das abordagens consistiu na combinação de Sistemas Fuzzy com a Computação Evolucionária, de forma a unir a interpretabilidade dos SIF com a capacidade de otimização dos Algoritmos Evolucionários, buscando uma maior acurácia [7, 11, 14].

Em 1991, três trabalhos pioneiros que usaram o Algoritmo Genético como meta-heurística genética (até então a única meta-heurística genética existente) abriram caminho para as diferentes abordagens até hoje usadas em SFE:

1. Seleção de parâmetros para funções de pertinência [15].
2. Busca de base de regras fuzzy, ao codificar em um indivíduo uma regra (abordagem Michigan) [16].
3. Busca de base de regras fuzzy tratando um indivíduo como uma base de regras (abordagem Pittsburgh) [17].

Alguns sistemas [13, 14] são denominados **Sistemas Fuzzy Evolucionários (SFE)** por serem considerados extensões dos tradicionais Sistemas Fuzzy-Genéticos. Nesta dissertação usar-se-á o termo **Sistemas Fuzzy Evolucionário** para representar a sinergia dos SIF com os Algoritmos Evolucionários; em [18] são descritas principais fontes.

2.1.1 Arquitetura básica

A arquitetura básica de um Sistema Fuzzy Evolucionário é o resultado da interação entre um Sistema de Inferência Fuzzy (SIF) e uma Meta-Heurística Evolucionária (MHE), conforme ilustrado na Figura 2.1.

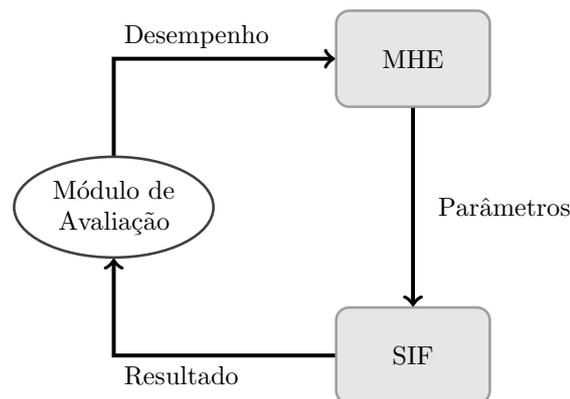


Figura 2.1: Diagrama genérico de um SFE

Define-se por MHE o tipo de Algoritmo Evolucionário usado para a construção de um SIF, como, por exemplo: Algoritmos Genéticos (AG), Programação Genética (PG), Algoritmos Evolucionários Multiobjetivos (MOEA pelas siglas em inglês). A MHE é usada como mecanismo de composição da base de regras e do ajuste de funções de pertinência, por exemplo, mas nunca como um ator que possa alterar o modo de inferência e a estrutura, como ocorre nos casos de Sistemas Neuro-Fuzzy [18].

Um Sistema de Inferência Fuzzy é composto por:

- Um mecanismo de Inferência Fuzzy.
- Base de Conhecimento (BC), que compreende:

- Base de Regras (BR)
- Base de Parâmetros (BP): fatores de escala do universo de discurso, tipo dos operadores de inferência e de defuzzificação, e, em especial, o número de funções de pertinência, seu formato e disposição.

O papel de um SFE é atuar em alguma área da BC, seja alterando a BP ou efetuando o aprendizado de uma BR. A Figura 2.2 ilustra o papel da MHE na atuação sobre o SIF.

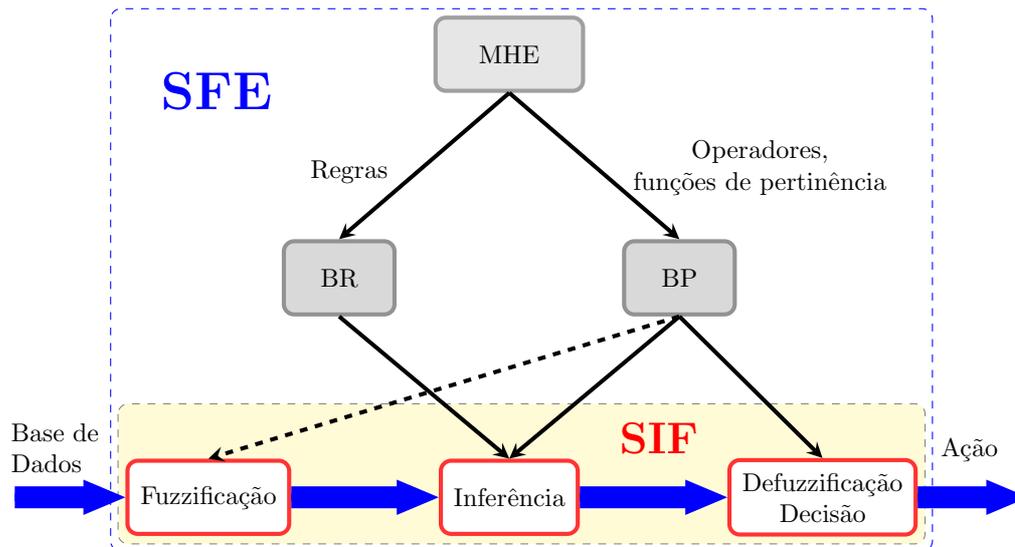


Figura 2.2: Atuação da MHE em um SFE, adaptada de [18].

2.1.2 Áreas de atuação

Em linhas gerais, um SFE atua em:

- **Ajuste Genético dos Parâmetros:** dada uma BC, a MHE ajusta os parâmetros da BP (funções de pertinência, operadores de t-norma, t-conorma e agregação) ou simplifica a BR (seleção de regras e aperfeiçoamento no modo de defuzzificação).
- **Descoberta de Conhecimento:** o SFE efetua o aprendizado completo de uma parte da BC. As principais linhas para a construção de um SFE são: aprendizado da granularidade (número e formato das funções de pertinência) e aprendizado da BR.

Conforme descrito em [18], os trabalhos mais recentes na literatura utilizam uma abordagem de tipo híbrida para tentar aproveitar as vantagens de cada tópico acima.

2.1.3 Abordagens para Base de Regras

A seguir são apresentadas as principais abordagens [7] para codificação de uma BR na MHE no âmbito dos SFEs: Pittsburgh, Michigan, Iterative Rule Learning (IRL) e Genetic Cooperative-Competitive Learning (GCCCL), conforme mostrado na Figura 2.3.

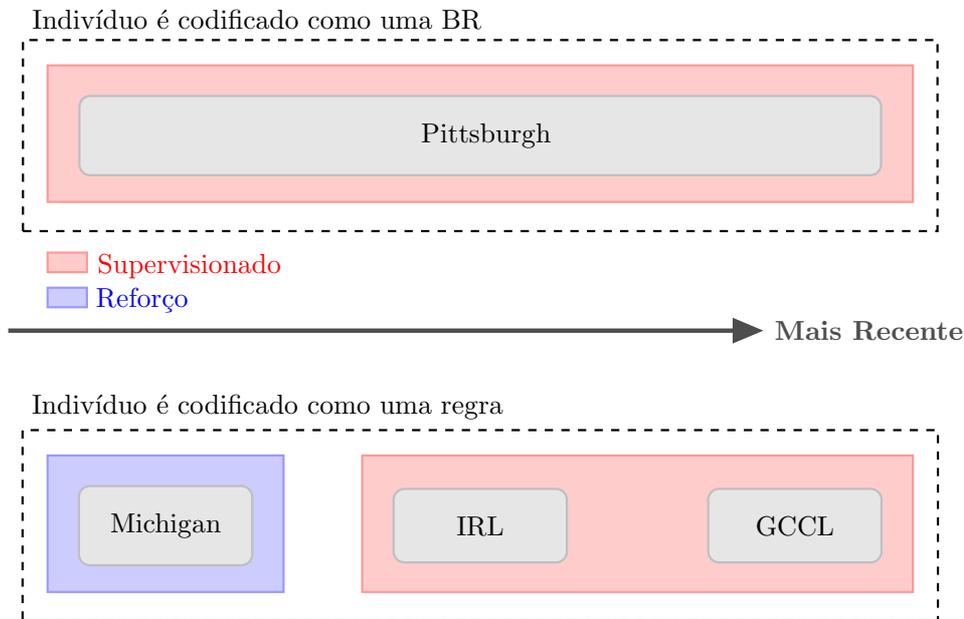


Figura 2.3: Abordagens de Bases de Regras de acordo a codificação do Indivíduo.

1. **Pittsburgh**: o usuário define uma BP fixa e o melhor indivíduo é aquele que satisfaz critérios tais como acurácia, interpretabilidade e tempo de resposta. O custo de concepção é baixo, pois basta definir uma função de avaliação;
2. **Michigan**: a BR final é resultado da concatenação dos indivíduos da MHE, o que reduz a sobrecarga computacional drasticamente [11]. Por outro lado, o procedimento de avaliação é mais complexo e indireto, já que uma regra pode representar bem o processo quando é isolada e nem tanto é inserida em uma BR;
3. **Iterative Rule Learning (IRL)**: executa-se a MHE um número de vezes e a melhor regra de cada execução é compõe a BR final;
4. **Genetic Cooperative-Competitive Learning (GCCL)**: abordagem híbrida, em que as regras competem e cooperam entre elas.

A abordagem do tipo Pittsburgh é prevalente em problemas de controle, reconhecimento de padrões e previsão. Por outro lado, as abordagens indivíduo-regra são principalmente voltadas para problemas de classificação [18], em

que as métricas mais comuns para avaliação de desempenho são o Grau de Confiança (*Confidence Degree - CD*) e o Suporte (*Sup*).

2.1.4 Aplicações em Classificação

- Em um dos primeiros trabalhos [19] – um SFE com abordagem tipo Pittsburgh, usando como MHE um AG –, verificava-se em todo o universo de possíveis regras quais seriam incluídas na BR. A definição da classe mais compatível com o termo antecedente foi introduzida e a maximização da acurácia e a redução do número de regras (fator de interpretabilidade) serviram como avaliação da qualidade da solução. Em [20], o trabalho foi aprimorado com a inclusão da MOEA para satisfazer ambos os critérios.
- Um dos trabalhos pioneiros na abordagem do tipo Michigan foi [16]. O modelo Fuzzy-XCS [21] apresenta uma roupagem moderna do método, com diversas possibilidades de uso.
- O modelo MOGUL [22] forneceu as bases para a abordagem do tipo IRL. Ressalta-se os modelos SLAVE [23], desenvolvido para atuar em bases de dados de elevada dimensionalidade, e 2SLAVE [24], que inclui no processo evolutivo um método de seleção de entradas. Em [25], há possibilidade de geração de regras relacionais fuzzy (modelagem conjunta dos termos antecedentes) e geração de novos atributos a partir de combinações lineares dos iniciais.
- A abordagem GCCL é apresentada em [26], que propõe um AG com uma representação de tamanho fixo para reconhecimento de padrões. O consequente e o peso da regra são determinados a partir do CD. Em [27], exibe-se uma aplicação da codificação GCCL para aprendizado de regras para um SIF do tipo TSK, onde a MHE usada é a PG com gramática livre de contexto.
- Em [28], um SFE hierárquico do tipo TSK é utilizado tanto para problemas de ajuste de curvas, assim como para classificação. Um AG realiza o aprendizado de toda a estrutura da BC, enquanto a parte consequente é realizada por mínimos quadrados.
- Em [29], realiza-se uma hibridização entre a abordagem Pittsburgh e o GCCL, de modo a aproveitar a maior acurácia da primeira, com o aumento de diversidade gerada pela GCCL.
- Em Fernández et al. [30], são explorados os modelos de González e Pérez [23] (tipo IRL) e Ishibuchi et al. [29] (híbrido Pittsburgh-GCCL),

de modo a investigar diferentes heurísticas de votação em problemas de classificação do tipo *pairwise learning* (um contra um).

- Recentemente, os modelos propostos têm se voltado para a redução de tempo computacional, introduzindo conceitos de computação paralela e novas formas de aprendizado [10], seleção de operadores de agregação mais adequados e SIF do tipo 2 com ajuste das funções de pertinência [31].

Os SFEs descritos acima baseiam-se ou em um MOEA ou no paradigma prevalente de AG. Ao se modificar a MHE de AG para PG, alguns autores auferiram ganhos em redução da complexidade da BR, assim como em acurácia final. Descrevem-se a seguir alguns trabalhos com a PG como MHE:

- Um dos primeiros trabalhos [32] faz uso da abordagem Pittsburgh. Apresentam-se diversos usos do modelo.
- Em [33], é proposto o modelo GP-COACH¹, do tipo GCCL para problemas com alta dimensionalidade. Realiza-se um estudo das vantagens do uso da PG em SFEBR com respeito a outras MHE. São utilizadas 24 bases de dados benchmarks para comparação com quatro SFEBR da literatura, baseados em diferentes codificações (GCCL, IRL e Pittsburgh). Suas principais características são:
 - Geração de regras do tipo *Disjunctive Normal Form* (DNF), do tipo “Se X_1 é A_1 ou A_2 e X_2 é A_3 ou A_4 e ...”, com uso de funções de pertinência triangulares nos termos linguísticos.
 - Promoção de diversidade para geração de regras (avaliação), geração de regras secundárias para cobrir padrões não classificados (recombinação) e competição entre os indivíduos da população anterior e da atual para permanecer na nova população (seleção).
- O modelo [33] é estendido em [34] para reconhecimento de padrões quando as classes são intensamente desbalanceadas.
- Em [35], é proposto o modelo FARC-HD para mineração de regras fuzzy em bases de dados de alta dimensionalidade. O modelo consiste nas seguintes etapas:
 - Extração de regras fuzzy: procuram-se as regras mais frequentes a partir da métrica do Suporte Modificado para contemplar o desbalanceamento entre as classes e assim poder gerar as regras candidatas.

¹Genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems

- Seleção de regras por qualidade de ajuste; a métrica denominada $wWRAcc$ é usada para avaliar a qualidade de cada regra candidata.
- Seleção de regras por AG e ajuste das funções de pertinência.
- Em [36], a PGMG², é usada para evoluir classificadores. Cada árvore do indivíduo representa todo um conjunto de regras associadas à h -ésima classe, ou seja, para um problema de H classes, é necessária a decodificação de H árvores no indivíduo. Em um trabalho similar [37], a decodificação de solução faz uso de operações matemáticas que se confundem com os operadores fuzzy.
- O modelo GPFIS [18] (Genetic Programming Fuzzy Inference System), inspirado em [33, 36, 37], surge como uma arquitetura genérica do tipo Pittsburgh que pode atuar em diferentes aplicações, e não somente em classificação. A definição da classe que melhor se ajusta ao antecedente da regra é realizada por uma heurística, prescindido de uma busca via PGMG. Permite a inclusão de vários tipos de t-norma, t-conorma, operadores de agregação, negação e modificadores linguísticos, de modo a maximizar a acurácia e a aceção linguística da solução.

2.2 Sistemas Fuzzy Evolutivos

Nestes sistemas, as regras fuzzy são construídas de forma gradual por um mecanismo de aprendizado. Este aprendizado de novas regras ocorre a partir de novos dados, preservando-se as regras já aprendidas. As vantagens de tais sistemas são flexibilidade, adaptação e robustez. De acordo com [38], os principais modelos dividem-se em:

- Modelos Fuzzy que usam um processo recursivo de clusterização, baseado no aprendizado participativo, para evoluir uma estrutura dinâmica (base de regras). São usadas funções de pertinência multivariadas para os conjuntos fuzzy dos antecedentes para evitar perda de informação nas interações das variáveis de entrada. O procedimento de clusterização recursivo é robusto para dados ruidosos e outliers.
- Modelos Fuzzy que usam uma árvore de regressão linear fuzzy como alternativa topológica e também como critério alternativo para evoluir a estrutura. Esta é atualizada por meio de testes estatísticos recursivos, substituindo-se folhas por subárvores que melhoram a qualidade do ajuste do modelo.

²Programação Genética Multigênica, a generalização da Programação Genética clássica

2.2.1 Sistema Evolutivo com Clusterização

Estes sistemas iniciaram-se aproximadamente nos inícios do ano 2000 e, como descrito em [13], não necessitam conhecer com antecedência a quantidade de regras a ser construídas, pois novas classes podem ser introduzidas durante o processo de aprendizagem.

Os principais algoritmos do tipo incremental, diferenciados pela arquitetura, são:

- **eClass0**: com estrutura do tipo Mamdani, alto nível de interpretação e cada classe é uma saída.

Regra_r: “Se X_1 é A_{l1} , X_2 é A_{l2}, \dots, X_J é A_{lJ} então \mathbf{x}_i é classe k ”

onde:

- *Regra_r* : i -ésima regra da base de R regras;
 - X_j : atributo j ;
 - \mathbf{x}_i : i -ésima instância da base de dados de n padrões;
 - A_{lj} : l -ésima função de pertinência do atributo j .
- **eClass1**: prevê a possibilidade de um vetor de características para uma certa classe; a estrutura é do tipo Takagi-Sugeno. Baseia-se na regressão sobre as características que produzem uma estimativa da possibilidade de que uma amostra de dados pode pertencer a certa classe.

Regra_r: “Se X_1 é A_{l1} , X_2 é A_{l2}, \dots, X_J é A_{lJ} , então
 $(p_r^1 = d_{r0}^1 + d_{r1}^1 X_1 + \dots + d_{rJ}^1 X_J)$ e
 \dots e $(p_i^K = d_{r0}^K + d_{r1}^K X_1 + \dots + d_{rJ}^K X_J)$ ”

onde:

- p_r^1 : r -ésimo vetor de características não normalizado para descrever a classe 1
- $\frac{p_r^1}{\sum_{j=1}^J p_r^j}$: r -ésimo vetor de características normalizado para descrever a classe 1
- K : número total de classes

- **Multi-Model EFS (eClassM)**: combinação de classificadores do tipo eClass1 (um para cada classe). Apresenta uma maior acurácia:

$$\text{Regra}_r^1: \text{“Se } X_1 \text{ é } A_{11}, X_2 \text{ é } A_{12}, \dots, X_J \text{ é } A_{1J} \\ \text{então}(p_r^1 = d_{r0}^1 + d_{r1}^1 x_j + \dots + d_{rJ}^1 X_J) \text{”}$$

$$\text{Regra}_r^2: \text{“Se } X_1 \text{ é } A_{11}, X_2 \text{ é } A_{12}, \dots, X_J \text{ é } A_{1J} \\ \text{então}(p_r^2 = d_{r0}^2 + d_{r1}^2 x_j + \dots + d_{rJ}^2 X_J) \text{”}$$

$$\text{Regra}_r^K: \text{“Se } X_1 \text{ é } A_{11}, X_2 \text{ é } A_{12}, \dots, X_J \text{ é } A_{1J} \\ \text{então}(p_r^K = d_{r0}^K + d_{r1}^K x_j + \dots + d_{rJ}^K X_J) \text{”}$$

Um algoritmo de clusterização é usado para adaptar a base de regras em cada iteração (quando uma nova amostra é apresentada) e os clusters são projetados em cada variável de entrada. Gera-se assim os antecedentes dos conjuntos fuzzy definidos por funções de pertinência Gaussianas [38] (Figura 2.4).

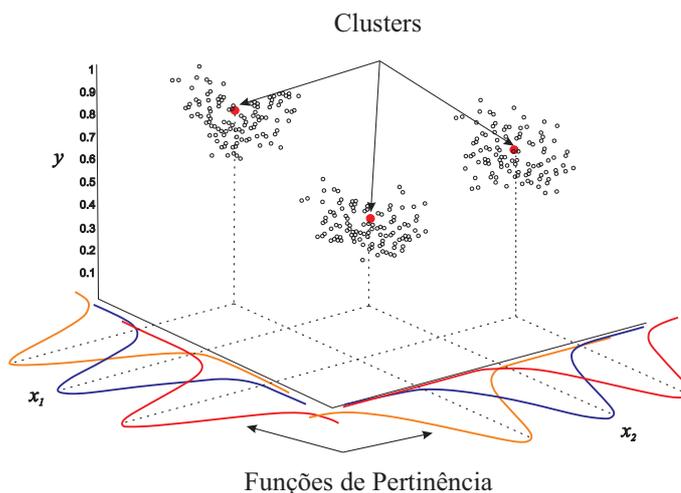


Figura 2.4: Conjuntos Fuzzy projetados pelos clusters, adaptada de [38]

Nesta abordagem, são identificados os protótipos (pontos focais das regiões com alta densidade de dados, que definem as regras) por meio de um fator denominado “potencial” (recursivo para novas instâncias) – medida da densidade de dados (por exemplo, a distância euclidiana) para uma classe específica. A Figura 2.5 mostra que o ponto A tem menor potencial do que o ponto B.

A função de pertinência que descreve o grau de associação de um protótipo específico tem forma gaussiana. Removem-se regras ambíguas após a inserção de uma nova regra. No **eClass1**, o potencial é calculado de forma global e não por classe (caso do **eClass0**), para assim identificar os protótipos representativos de qualquer classe.

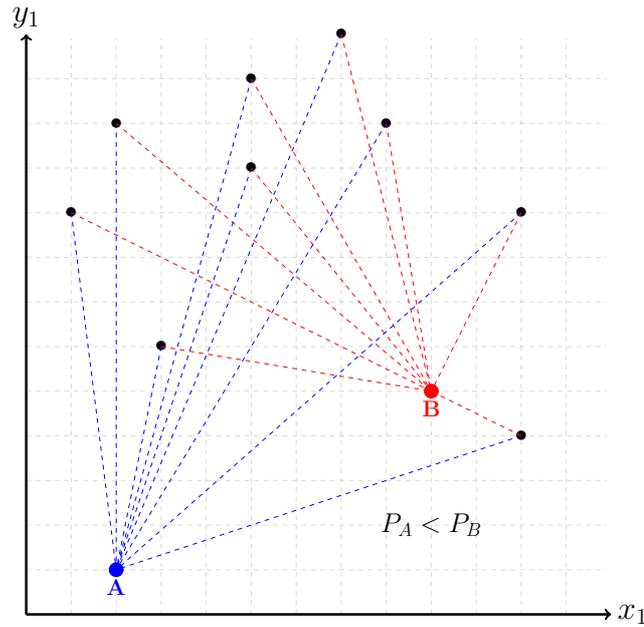


Figura 2.5: Ilustração do potencial das amostras de dados, adaptada de [38]

2.2.2 Sistema Evolutivo com Árvore de Regressão Linear Fuzzy

As árvores de regressão linear fuzzy são uma generalização das árvores de regressão (Figura 2.6), em que cada folha representa uma região do espaço de entrada. O espaço é delimitado pelos pontos de divisão dos nós internos no caminho que se inicia na raiz (Figura 2.7.a) e termina nas folhas. A introdução de funções de pertinência sigmoidais nas folhas criam regiões com fronteiras fuzzy (Figura 2.7.b)

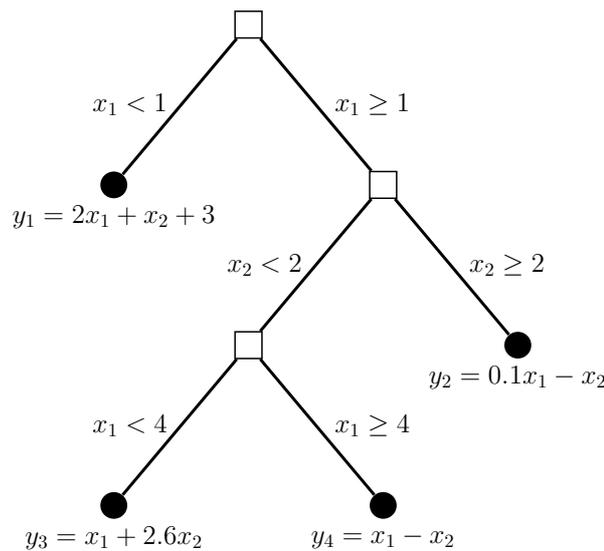


Figura 2.6: Exemplo de árvore de regressão linear, adaptada de [38]

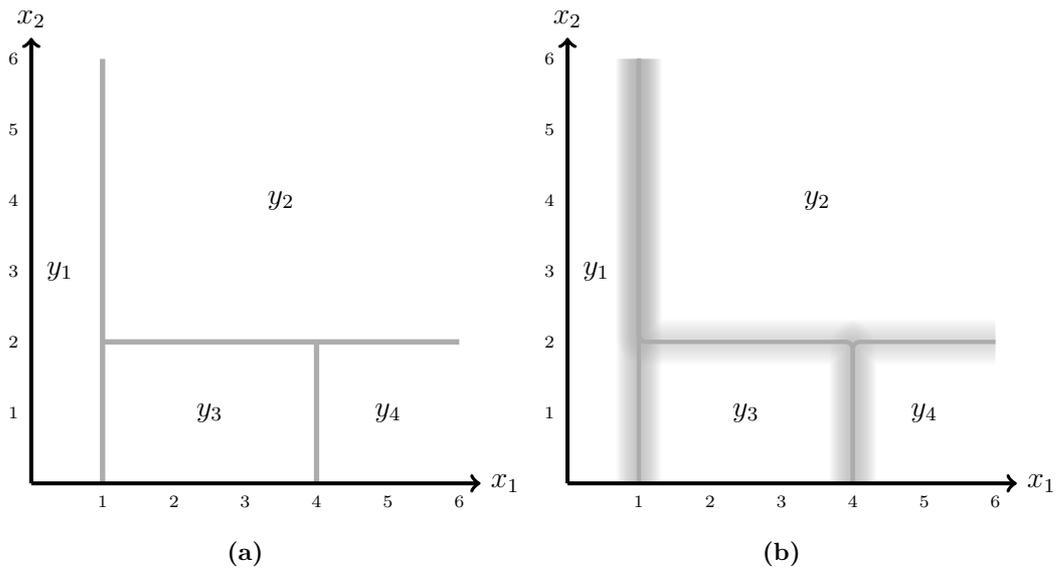


Figura 2.7: (a) Partição do espaço de entrada de uma árvore de regressão linear. (b) Partição do espaço de uma árvore de regressão linear fuzzy, adaptada de [38]

No crescimento da árvore, supõe-se que cada folha tenha, por exemplo, P divisões candidatas para cada J atributos, totalizando assim $P \times J$ possíveis divisões. Qualquer delas pode substituir uma folha existente. Assim, a estrutura é atualizada via um teste de seleção estatística de forma recursiva, substituindo folhas por subárvores que melhoram a qualidade do ajuste do modelo resultante.

2.3 Interpretabilidade

Em geral, o desenvolvimento de um SIF apresenta dois requisitos: a interpretabilidade e a acurácia. Esta é expressa objetivamente; sua medição é direta e bem conhecida. Quanto à interpretabilidade, é difícil defini-la e medi-la, não apenas pelo número de fatores envolvidos, mas também devido a sua natureza subjetiva [8,14]. Por isso as medidas de interpretabilidade ainda são um problema em aberto. Nos últimos anos, esse tópico tem ganhado a atenção, não só por razões acadêmicas, mas também porque muitas aplicações requerem modelos mais compreensíveis para explicar um determinado fenômeno. Entre as recentes e principais taxonomias propostas tem-se:

- Alonso [3]: apresenta os principais fatores a serem considerados para avaliar a interpretabilidade dos Sistemas de Inferência Fuzzy baseados em regras, agrupando-os em **legibilidade** e **compreensão do sistema (explicação)**. A **legibilidade** contém fatores como Base de Regras, Proposições Linguísticas, Particionamentos Linguísticos, Particionamentos Fuzzy e Conjuntos Fuzzy. Em relação à **explicação**, os fatores incluem os operadores de disjunção e conjunção, defuzzificação, consistência, integridade, negação, número de variáveis, estrutura do modelo (Takagi-Sugeno, Mamdani), etc.
- Mencar [39]: analisa as restrições nos tópicos de conjuntos fuzzy, universo de discurso, aprendizado da granularidade, regras fuzzy, modelos fuzzy e algoritmos de aprendizagem. Explica a importância das restrições mencionadas no processo de granulação, pois assim a interpretabilidade pode ser garantida.
- Zhou e Gan [40]: propõe uma taxonomia em termos de interpretabilidade de baixo e alto nível. No baixo nível (nível dos conjuntos fuzzy), funções de pertinência são otimizadas em termos de critérios semânticos – Interpretabilidade baseada na Semântica. No alto nível (nível das regras fuzzy), busca-se reduzir a complexidade pela obtenção de um número moderado de variáveis e regras – Interpretabilidade baseada na Complexidade. Além disso, outro critério no alto nível é a coerência das regras, ou seja, a interpretabilidade linguística também está presente neste nível.
- Gacto2011 [8]: propõe o modelo dos quatro quadrantes definidos por dois tipos de medidas que influenciam a interpretabilidade (Complexidade e Semântica) nos componentes da BC (particionamento Fuzzy e a BR). A primeira medida é dedicada a diminuir a complexidade do modelo obtido; a segunda, a preservar a semântica associada às funções de pertinência.

O trabalho aqui desenvolvido concentra-se na taxonomia proposta em [8]. Definem-se, na Figura 2.8, os quatro quadrantes³ mencionados anteriormente:

- Q-I: Complexidade no nível da base de regras
- Q-II: Complexidade no nível do particionamento fuzzy
- Q-III: Semântica no nível do particionamento fuzzy
- Q-IV: Semântica no nível da base de regras

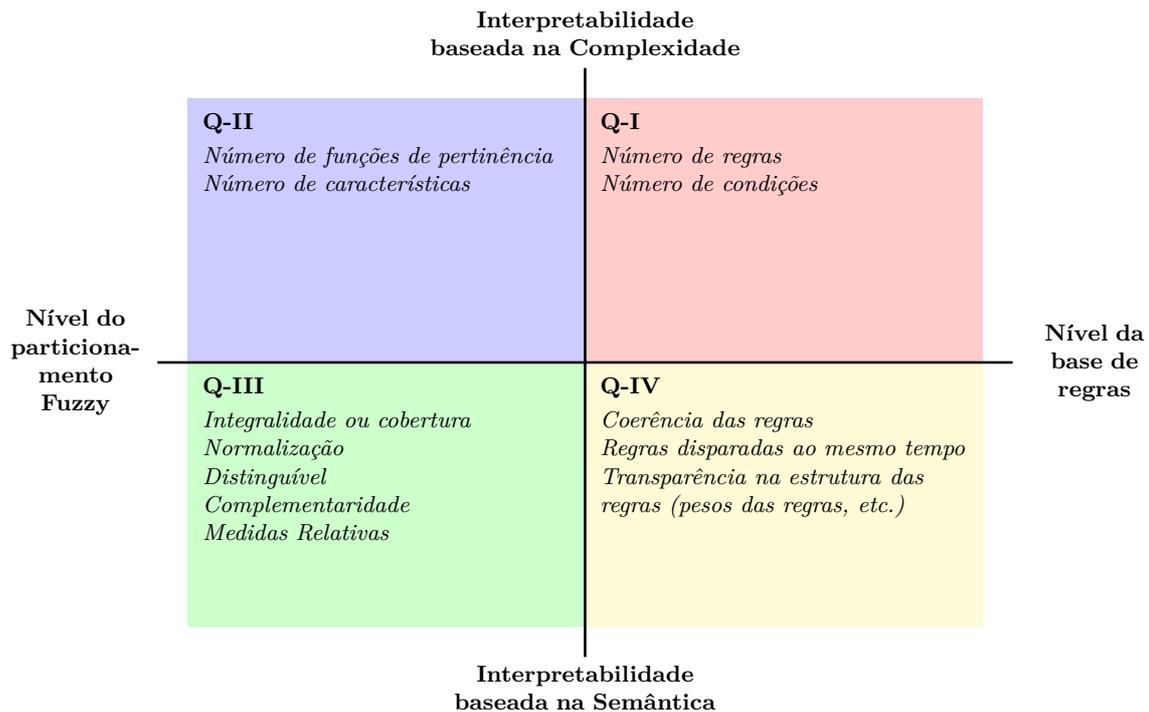


Figura 2.8: Indicadores nos Quadrantes de Interpretabilidade

Os indicadores dos quadrantes da Interpretabilidade são explicados na Tabela 2.1. A maioria dos trabalhos presentes na literatura concentra-se em Q-I e Q-II. Os poucos trabalhos em Q-I, tratam preponderantemente da Consistência de Base de Regras e da aplicação desta propriedade em função do tipo de problema (classificação, regressão, controle, etc). O número de regras ativadas ao mesmo tempo é outro indicador promissor.

³A correspondência com o artigo original do autor difere em 2 quadrantes, onde Q-IV é igual a Q3 e Q-III é igual a Q4

		Indicadores
Q-I	Número de regras	Evoca o princípio lógico da Navalha de Occam : “Diante de várias explicações para um problema, a mais simples tende a ser a mais correta”. Nosso contexto dos SIF, o conjunto de regras fuzzy deve ser o menor possível, preservando-se em um nível satisfatório o desempenho do modelo.
	Número de condições no antecedente	Não deve exceder o limite de 7 ± 2 condições distintas, número de entidades conceituais com as quais um ser humano consegue lidar.
Q-II	Número de funções de pertinência	A redução na dimensionalidade (características) poder melhorar a legibilidade dos SIF.
	Número de características	A redução na dimensionalidade (características) poder melhorar a legibilidade dos SIF.
Q-III	Integralidade ou Cobertura	Cada ponto deve ser coberto ao menos por um termo linguístico
	Normalização	Se houver pelo menos um ponto no universo do discurso com um valor de pertinência igual a um.
	Distinguíbilidade	Uma função de pertinência deve representar um termo linguístico com um significado semântico claro e distinguível das restantes funções de pertinência da variável correspondente.
	Complementaridade	Para cada elemento do universo do discurso, a soma de todos os seus valores de adesão deve ser próxima de um. Isto garante uma distribuição uniforme dos significados entre os elementos.
Q-IV	Consistência da base de regras	Ausência de regras contraditórias: premissas similares com consequentes diferentes.
	Número de regras ativadas no mesmo tempo	Procura-se minimizar as regras ativadas para uma dada entrada.

Tabela 2.1: Indicadores dos Quadrantes de Interpretabilidade

3

Modelo Automatic Fuzzy Inference System (AutoFIS)

Esse capítulo apresenta o modelo proposto Automatic Fuzzy Inference System (AutoFIS), uma nova variante dos classificadores fuzzy com o uso de uma abordagem tipo *off-line*. Neste modelo, procura-se uma boa acurácia aliada à interpretabilidade linguística.

Para garantir uma boa interpretabilidade, o modelo constrói as regras com restrições tais como:

- Limitação do tamanho máximo das premissas.
- Avaliação da viabilidade através de um conjunto de filtros.
- Geração expansiva e organizada das premissas, iniciando pelas de tamanho 1 e sucessivamente aumentando o tamanho delas de forma gradual.

As etapas de construção, após a escolha dos parâmetros do modelo, são:

1. Fuzzificação
2. Formulação das premissas.
3. Associação das premissas aos consequentes.
4. Agregação das premissas de cada consequente.
5. Decisão
6. Avaliação

3.1 Definições e Notações

Em classificação, a principal informação disponível consiste nos n padrões $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iJ}]$, compostos de J atributos X_j presentes na base de dados ($i = 1, \dots, n$ e $j = 1, \dots, J$), conforme apresentado na Tabela 3.1.

Padrão	Atributos						Saída
	X_1	X_2	...	X_j	...	X_J	
1	x_{11}	x_{12}	...	x_{1j}	...	x_{1J}	\vdots
2	x_{21}	x_{22}	...	x_{2j}	...	x_{2J}	\vdots
3	x_{31}	x_{32}	...	x_{3j}	...	x_{3J}	\vdots
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	C_k
i	x_{i1}	x_{i2}	...	x_{ij}	...	x_{iJ}	\vdots
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	C_k
n	x_{n1}	x_{n2}	...	x_{nj}	...	x_{nJ}	\vdots

Tabela 3.1: Representação de uma base de dados

A cada j -ésimo atributo são associados L conjuntos fuzzy $A_{jl} = \{(x_{ij}, \mu_{A_{jl}}(x_{ij})) | x_{ij} \in X_j\}$, em que $\mu_{A_{jl}} : X_j \rightarrow [0, 1]$ é uma função de pertinência que associa a cada observação x_{ij} um grau de pertinência (compatibilidade) $\mu_{A_{jl}}(x_{ij})$ ao conjunto fuzzy A_{jl} (Tabela 3.2). Por fim, cada padrão está associado a uma classe C_k das K possíveis, isto é, $C_k \in \{1, 2, \dots, K\}$.

Padrão	Atributos				Conjuntos Fuzzy associados					
	...	X_j	X_j					
1	...	x_{1j}	$u_{A_{j1}}(x_{1j})$...	$u_{A_{jl}}(x_{1j})$...	$u_{A_{jL}}(x_{1j})$...
2	...	x_{2j}	$u_{A_{j1}}(x_{2j})$...	$u_{A_{jl}}(x_{2j})$...	$u_{A_{jL}}(x_{2j})$...
3	...	x_{3j}	$u_{A_{j1}}(x_{3j})$...	$u_{A_{jl}}(x_{3j})$...	$u_{A_{jL}}(x_{3j})$...
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
i	...	x_{ij}	$u_{A_{j1}}(x_{ij})$...	$u_{A_{jl}}(x_{ij})$...	$u_{A_{jL}}(x_{ij})$...
\vdots	\vdots	\vdots	\vdots
n	...	x_{nj}	$u_{A_{j1}}(x_{nj})$...	$u_{A_{jl}}(x_{nj})$...	$u_{A_{jL}}(x_{nj})$...

Tabela 3.2: Associação de funções de pertinência a um atributo

3.2 Arquitetura

Apresenta-se abaixo o pseudocódigo das etapas do modelo AutoFIS-CLASS, destinado especificamente a tarefas de classificação. Premissas são representadas por P ; P_n indica premissas de tamanho n .

AutoFIS

Parâmetros do modelo:

- 1: $X \leftarrow$ base de dados
- 2: $N \leftarrow$ tamanho máximo das premissas
- 3: $L \leftarrow$ número de funções de pertinência
- 4: $t_{Fuz} \leftarrow$ tipo de formatação triangular
- 5: $t_{Sup} \leftarrow$ tipo de Filtro de Suporte
- 6: $\varepsilon_{Sup} \leftarrow$ limiar de Filtro de Suporte
- 7: $\varepsilon_{Sim} \leftarrow$ limiar de Similaridade
- 8: $e_{neg} \leftarrow$ habilitar negação
- 9: $m_{assoc} \leftarrow$ método de associação
- 10: $m_{agreg} \leftarrow$ método de agregação

Fuzzificação:

- 11: $u_X \leftarrow$ *fuzzificar*(X, L, t_{Fuz}, e_{neg})

Formulação:

- 12: $P_{individuais} \leftarrow$ *GerarPremisas*(u_X)
- 13: $P_1 \leftarrow$ *Filtros*($P_{individuais}, u_X, t_{Sup}, \varepsilon_{Sup}, \varepsilon_{Sim}$)
- 14: **for** $n = 2$ to N **do**
- 15: $P_{candidatas} \leftarrow$ *Expandir*(P_{n-1})
- 16: $P_n \leftarrow$ *Filtros*($P_{candidatas}, u_X, t_{Sup}, \varepsilon_{Sup}, \varepsilon_{Sim}$)
- 17: **if** $P_n = \emptyset$ **then**
- 18: **break**
- 19: **end if**
- 20: **end for**

Associação:

- 21: $P_{classes} \leftarrow$ *Associar*(P, m_{assoc})

Agregação:

- 22: $PremisasFinais_{classes} \leftarrow$ *Agregar*($P_{classes}, m_{agreg}$)
-

O processo de construção do modelo inicia-se pela Fuzzificação. Em seguida, o processo passa para a elaboração das premissas de regras de forma gradual (Formulação), a definição do conseqüente mais apropriado (Associação) e a união das ativações de cada regra (Agregação). A classe vencedora para um determinado padrão é definida na etapa de Decisão.

As linhas gerais de construção do modelo, após a etapa de Fuzzificação, são mostradas na Figura 3.1.

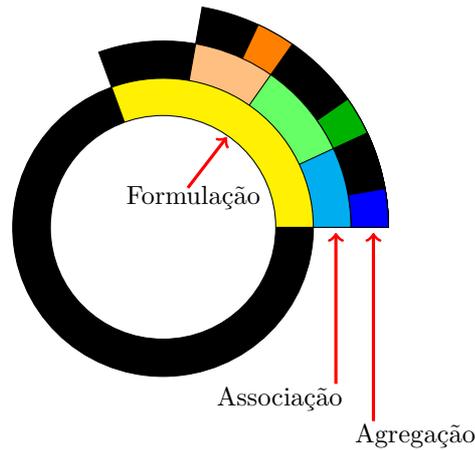


Figura 3.1: Etapa de Inferência no modelo AutoFIS-Class.

A coroa circular interna simboliza o processo de Formulação. A região escura representa todas as premissas elaboradas, ao passo que a faixa amarela corresponde àquelas que satisfizeram os critérios mínimos de qualidade para passar à etapa de Associação. Nesta etapa, separam-se as premissas para diferentes classes (por isso as várias cores), sendo que algumas não são úteis a classe alguma (região em preto). Por fim, a etapa de Agregação, ao ponderar as regras, exclui aquelas (representadas pelas regiões em preto) com pouca influência no processo de classificação.

3.2.1 Fuzzificação

A etapa de Fuzzificação leva em conta três fatores: a forma das funções de pertinência, o suporte de cada função de pertinência $\mu_{A_{jl}}(x_{ij})$ e o rótulo linguístico apropriado, qualificando o subespaço compreendido pela função de pertinência com um adjetivo correspondente ao contexto. Em teoria, essas tarefas devem ser desempenhadas por um especialista no assunto, proporcionando ganhos de interpretabilidade nas regras fuzzy geradas. Na prática, entretanto, devido à dificuldade de se encontrar um profissional capaz de qualificar o problema (em alguns casos ele não existe), é bastante comum [10,18,33] empregar uma disposição dita Fortemente Particionada, conforme mostrado na Figura 3.2(a). Outra abordagem faz uso da informação dos quartis: 0º quartil (va-

lor mínimo), 1º quartil, 2º quartil (mediana), 3º quartil e 4º quartil (valor máximo). Este tipo de particionamento é denominado Tukey e dá origem a disposições do tipo mostrado na Figura 3.2(b).

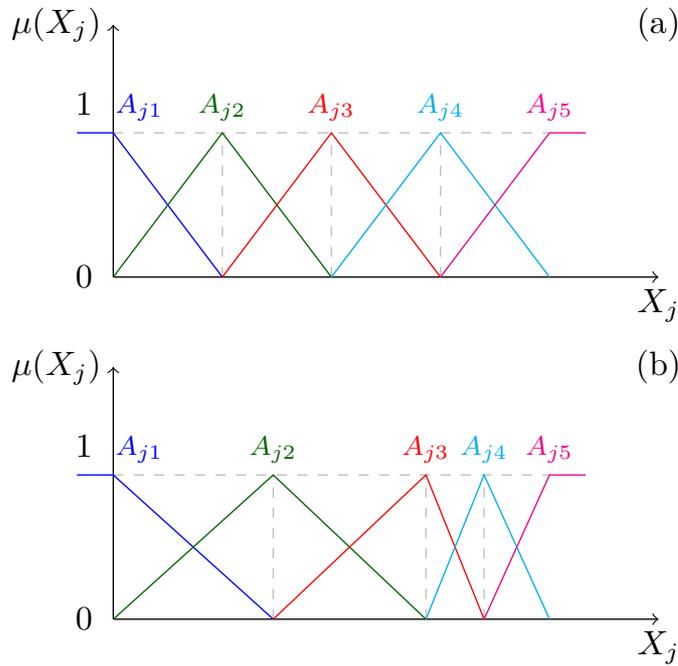


Figura 3.2: Tipo de funções de pertinência triangular: (a) Fortemente Particionada e (b) Tukey.

Ressalte-se que, neste trabalho, os atributos categóricos e as classes são binarizadas. A Tabela 3.3 ilustra um exemplo de uma base de dados com 3 atributos, sendo um deles do tipo categórico (Atributo 2). A correspondente transformação dessa base de dados, após a binarização do atributo categórico e da classe de saída, é apresentada na Tabela 3.4.

Padrão	Atributos		Saída
	A_1	A_2	C
\mathbf{x}_1	x_{11}	x_{12}	C_1
\mathbf{x}_2	x_{21}	x_{22}	C_2
\vdots	\vdots	\vdots	\vdots
\mathbf{x}_n	x_{n1}	x_{n2}	C_3

Tabela 3.3: Base de dados de Classificação ($k \in \{1, 2, 3\}$)

Padrão	Atributo 1		Atributo 2				Saída Bin		
	A_{1L}	A_{1H}	A_{21}	A_{22}	A_{23}	A_{24}	C_1	C_2	C_3
\mathbf{x}_1	$u_{A_{1L}}(x_{11})$	$u_{A_{1H}}(x_{11})$	1	0	0	0	1	0	0
\mathbf{x}_2	$u_{A_{1L}}(x_{21})$	$u_{A_{1H}}(x_{21})$	0	0	1	0	0	1	0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
\mathbf{x}_n	$u_{A_{1L}}(x_{n1})$	$u_{A_{1H}}(x_{n1})$	0	0	0	1	0	0	1

Tabela 3.4: Fuzzificação dos atributos (não categóricos) e Binarização dos atributos categóricos e saída.

Tanto a abordagem Fortemente Particionada quanto a abordagem Tukey evitam uma baixa interpretabilidade semântica no nível da partição fuzzy [8]. Além disso, com objetivo de manter uma boa interpretabilidade, o número de termos linguísticos (granularidade) por variável é restrito a 3, 5 e 7.

3.2.2 Formulação

Esta etapa consiste na elaboração das premissas de regras fuzzy. Uma premissa é comumente definida por: “Se X_1 é A_{1l} e ... e X_j é A_{jl} e ... e X_J é A_{Jl} ” ou, em termos matemáticos, como:

$$\mu_{A_d}(\mathbf{x}_i) = \mu_{A_{1l}}(x_{i1}) * \dots * \mu_{A_{jl}}(x_{ij}) * \dots * \mu_{A_{Jl}}(x_{iJ}) \quad (3-1)$$

onde $\mu_{A_d}(\mathbf{x}_i)$ é o grau de pertinência conjunto do i -ésimo padrão na d -ésima premissa ($d = 1, \dots, D$), computado geralmente a partir de uma t-norma * que combina cada $\mu_{A_{jl}}(x_{ij})$. De forma mais genérica, uma premissa pode ser construída a partir de uma combinação das $\mu_{A_{jl}}(x_{ij})$ com o uso de t-normas, t-conormas, operadores de negação e modificadores linguísticos.

Neste trabalho, considera-se apenas a t-norma produto. Além disso, o operador de negação pode incidir sobre cada elemento que compõe uma premissa: “Se X_1 Não é A_{1l} e ... e X_j Não é A_{jl} e ... e X_J Não é A_{Jl} ”, em oposição ao procedimento de negar o resultado como um todo:

$$[\text{Não}(\text{“Se } X_1 \text{ é } A_{1l} \text{ e } \dots \text{ e } X_j \text{ é } A_{jl} \text{ e } \dots \text{ e } X_J \text{ é } A_{Jl}\text{”})].$$

Apesar do emprego restrito da t-norma produto e do operador de negação por elemento, o número de possíveis premissas para formar uma base de regras fuzzy cresce de maneira exponencial à medida que mais atributos são adicionados na base de dados. A título de exemplo, considere-se um problema com três atributos (X_1, X_2, X_3) e que a cada um deles sejam associados dois termos linguísticos (Figura 3.3). Têm-se como premissas: de tamanho (número de elementos antecedentes) 1 “Se X_1 é A_{1L} ”, “Se X_1 é A_{1H} ”, ..., e “Se X_3 é A_{3H} ”; de tamanho 2 “Se X_1 é A_{1L} e X_2 é A_{2L} ”, ..., e “Se X_3 é A_{3H} e X_1 é A_{1H} ”;

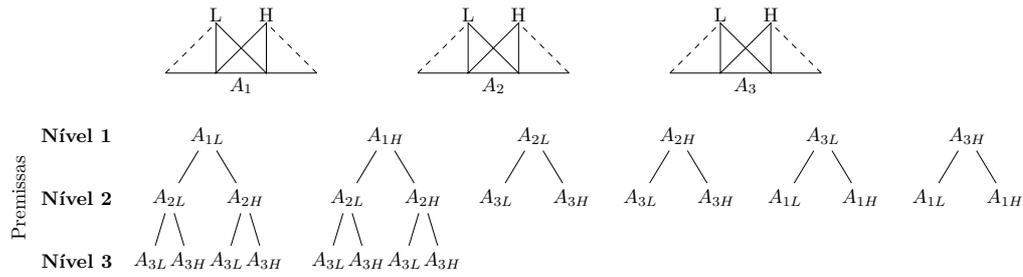


Figura 3.3: Premissas agrupadas pelo tamanho das premissas para 3 atributos com 2 conjuntos fuzzy cada um.

e de tamanho 3 “Se X_1 é A_{1L} e X_2 é A_{2L} e X_3 é A_{3L} ”, ..., e “Se X_1 é A_{1H} e X_2 é A_{2H} e X_3 é A_{3H} ”. No total, são seis premissas de tamanho 1, doze de tamanho 2 e oito de tamanho 3. Obviamente, quando se eleva o número de atributos e termos linguísticos, gera-se conseqüentemente uma quantidade imensa de premissas. Contudo, nem todas as premissas serão usadas para constituir uma base de regras, pois muitas delas ou são redundantes ou geram conflitos na decisão da classe mais apropriada.

Do ponto de vista da interpretabilidade, é desejável que se tenham poucas regras e que estas contenham poucos elementos antecedentes. Assim, propõe-se o seguinte procedimento para a geração de premissas:

1. Limitação no tamanho máximo de premissas;
2. Avaliação da viabilidade de uma premissa a partir de um conjunto de filtros: suporte, similaridade e grau de conflito na classificação;
3. Geração de premissas em um esquema do *menor para o maior não exaustivo*, isto é, criação inicialmente, de premissas de tamanho 1, criação das premissas de tamanho 2 a partir das viáveis de tamanho 1, criação das premissas de tamanho 3 a partir das viáveis de tamanho 2, e assim sucessivamente.

Com respeito ao primeiro item acima, o tamanho máximo das premissas é um parâmetro definido pelo usuário; o seu valor máximo é o número de atributos presentes na base de dados. Quanto ao terceiro item, o objetivo é gerar premissas com poucos elementos antecedentes (mais interpretáveis), assim como não incorrer em uma sobrecarga computacional com premissas não apropriadas para a formação de uma regra fuzzy (cf. resultado do item 2). Descreve-se a seguir o procedimento de filtragem (item 2).

Filtro do Suporte O filtro do suporte visa à elaboração de premissas que possam cobrir um número elevado de padrões na base de dados. Dada uma premissa $\mu_{A_d}(\mathbf{x}_i)$, o seu *Suporte Relativo* é dado por:

$$Sup_d = \frac{\sum_{i=1}^n \mu_{A_d}(\mathbf{x}_i)}{n} \quad (3-2)$$

Dada uma tolerância ε_{sup} definida pelo usuário, uma premissa é viável se $Sup_d > \varepsilon_{sup}$. Caso $\mu_{A_*}(\mathbf{x}_i) = \mu_{A_{1L}}(x_{i1})$ não obtenha *Suporte Relativo* maior do que ε_{sup} , qualquer combinação de $\mu_{A_*}(\mathbf{x}_i)$ com outra função de pertinência irá gerar premissas inviáveis. Isto pode ser verificado por meio da propriedade de degeneração ou não idempotência da maior parte das t-normas [41]. Tal observação é ilustrada na Figura 3.4 e Figura 3.5.

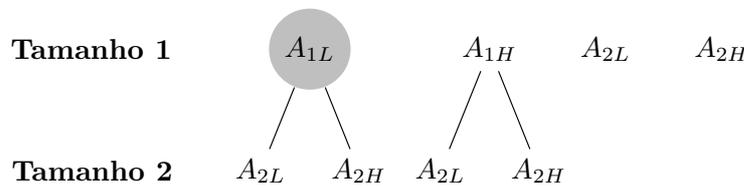


Figura 3.4: Premissa que não excede o limiar do Suporte

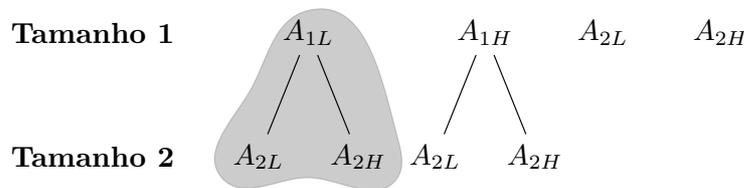


Figura 3.5: Premissas dispensadas do espaço de busca

Define-se a *Frequência Relativa*($freq_{rel}$) como:

$$freq_{relativa} = \frac{\sum_{i=1}^n I_i^{\{0,1\}}}{n}, I_i^{\{0,1\}} = \begin{cases} 1 & , \text{se } \mu_{A_d}(\mathbf{x}_i) > 0 \\ 0 & , \text{se } \mu_{A_d}(\mathbf{x}_i) = 0 \end{cases} \quad (3-3)$$

A *Frequência Relativa* é uma variação do *Suporte Relativo*, aumentando a possibilidade de que premissas que cobrem classes com poucas instâncias não sejam desconsideradas tão facilmente ao serem comparadas com a tolerância ε_{sup} definida pelo usuário.

Para ilustrar os conceitos, a Tabela 3.5 apresenta duas premissas A_{11} e A_{32} e suas duas possíveis premissas derivadas com o uso das t-normas mínimo e produto; a Tabela 3.6 mostra o *Suporte Relativo* e a *Frequência Relativa* para as quatro premissas considerando o valor de ε_{sup} igual a 0.10.

A_{11}	A_{32}	$prod(A_{11}, A_{32})$	$min(A_{11}, A_{32})$
0.0200	0.2500	0.0050	0.0200
0.0050	0.2000	0.0010	0.0050
0.0200	0.0300	0.0006	0.0200
0.0030	0.1000	0.0003	0.0030
0.0350	0.0000	0.0000	0.0000
0.0120	0.1000	0.0012	0.0120

Tabela 3.5: Premissas para analisar o efeito da Frequência Relativa

	A_{11}	A_{32}	$prod(A_{11}, A_{32})$	$min(A_{11}, A_{32})$
Suporte Relativo	0.0158	0.2500	0.0013	0.0133
Frequência Relativa	1.000	0.8333	0.8333	0.8333

Tabela 3.6: Efeito da diferença entre o Suporte Relativo e a Frequência Relativa

No caso do *Suporte Relativo*, só uma premissa (A_{32}) logrou superar o limiar definido ($\varepsilon_{Sup} = 0.10$); já no caso da *Frequência Relativa*, todas as quatro premissas superaram o limiar. Observa-se que as premissas $prod(A_{11}, A_{32})$ e $min(A_{11}, A_{32})$ no *Suporte Relativo* não conseguem superar o ε_{Sup} devido a A_{11} , ilustrando assim o efeito da não idempotência das t-norma mínimo e produto.

Filtro de Similaridade O objetivo deste filtro é reduzir a ocorrência de premissas semelhantes ou iguais. Dadas duas premissas $\mu_{A_d}(\mathbf{x}_i)$ e $\mu_{A_v}(\mathbf{x}_i)$, a similaridade entre elas pode ser medida como:

$$Sim_{d,v} = \frac{\sum_{i=1}^n \min\{\mu_{A_d}(\mathbf{x}_i), \mu_{A_v}(\mathbf{x}_i)\}}{\sum_{i=1}^n \max\{\mu_{A_d}(\mathbf{x}_i), \mu_{A_v}(\mathbf{x}_i)\}} \quad (3-4)$$

Dada uma tolerância ε_{sim} definida pelo usuário, duas premissas são similares se $Sim_{d,v} > \varepsilon_{sim}$. Se for identificada a similaridade, remove-se a premissa com o menor Suporte Relativo. Observe-se que, se uma premissa for excluída quando $Sim_{d,v} > \varepsilon_{sim}$, não necessariamente haverá a exclusão de uma combinação dela com outra função de pertinência (isto é, uma premissa maior derivada dessa). Contudo, opta-se por excluí-la do processo de forma a gerar o menor número possível de premissas para os demais passos pós-formulação.

Filtro PCD Este filtro busca reduzir a ocorrência de regras conflitantes. Para tanto, calcula-se o Grau de Confiança Penalizado [30] - PCD_k -, dado por:

$$PCD_k = \max \left\{ \frac{\sum_{i \in k} \mu_{A_d}(\mathbf{x}_i) - \sum_{i \notin k} \mu_{A_d}(\mathbf{x}_i)}{\sum_{i=1}^n \mu_{A_d}(\mathbf{x}_i)}, 0 \right\} \quad (3-5)$$

Assim, buscam-se regras que são mais específicas a uma determinada classe e não as mais generalistas. Somente são viáveis as premissas com $PCD_k > 0$. Após a geração de D premissas $\mu_{A_1}(\mathbf{x}_i)$, $\mu_{A_2}(\mathbf{x}_i)$..., $\mu_{A_D}(\mathbf{x}_i)$ que atendam aos limites de tamanho e que passem por todos os filtros, o próximo passo é associar cada premissa a uma classe consequente.

3.2.3 Associação

Determina-se, nesta etapa, a classe consequente mais compatível com uma dada premissa $\mu_{A_d}(\mathbf{x}_i)$. A d -ésima premissa associada à classe k (isto é, uma regra fuzzy) é denotada por $\mu_{A_{d(k)}}(\mathbf{x}_i)$, que descreve, em termos linguísticos:

“Se X_1 é A_{1l} , e ..., e X_J é A_{Jl} , então \mathbf{x}_i é Classe k ”

Dentre as diversas possibilidades de associação, o AutoFIS emprega:

- Grau de Confiança
- Grau de Confiança Penalizado
- Mínimos Quadrados Restritos

3.2.3.1 Grau de Confiança

O Grau de Confiança (CD_k) [29, 33] é definido por:

$$CD_k = \frac{\sum_{i \in k} \mu_{A_d}(\mathbf{x}_i)}{\sum_{i=1}^n \mu_{A_d}(\mathbf{x}_i)} \in [0, 1] \quad (3-6)$$

onde a parcela $\sum_{i \in k} \mu_{A_d}(\mathbf{x}_i)$ é a soma do grau de compatibilidade da premissa com a classe k e $\sum_{i=1}^n \mu_{A_d}(\mathbf{x}_i)$ é a compatibilidade total. A classe a ser definida para a $\mu_{A_d}(\mathbf{x}_i)$, isto é, $\mu_{A_{d(k)}}(\mathbf{x}_i)$, é dada pela k -ésima classe que maximizar CD_k . Caso alguma premissa possua $CD_k = 0$, ela não é associada a classe consequente alguma.

3.2.3.2 Grau de Confiança Penalizado

O Grau de Confiança Penalizado (PCD_k) tem como objetivo contabilizar o grau de ativação das demais classes e descontá-las do CD_k . Assim, buscam-se regras que são mais específicas a uma determinada classe e não as mais generalistas.

A definição da classe consequente é dada pelos passos abaixo, tal que, para todo o $\mu_{A_d}(\mathbf{x}_i)$:

1. Calcula-se o CD_k ou PCD_k para todas as K classes.
2. A classe consequente do $\mu_{A_d}(\mathbf{x}_i)$ é determinada pela que obtiver maior CD_k ou PCD_k .
3. Caso algum $\mu_{A_d}(\mathbf{x}_i)$ tenha $CD_k = 0$ ou $PCD_k = 0$ para todo k , esta é excluída. É possível observar que nem toda $\mu_{A_d}(\mathbf{x}_i)$ deverá ter um consequente, e ainda, não é garantido que toda a classe possa ter uma $\mu_{A_d}(\mathbf{x}_i)$ associada.

Para ilustrar esses conceitos, a Figura 3.6 expõe a relação de cinco premissas (P_1, P_2, P_3, P_4 e P_5) com três classes (C_1, C_2 e C_3) mediante o Grau de Confiança (CD). A mesma figura apresenta o Grau de Confiança Penalizado (PCD), baseado na equação 3-5.

	C_1	C_2	C_3
P_1	0	0.8	0.2
P_2	0.4	0.2	0.4
P_3	0.7	0.3	0
P_4	0.3	0.4	0.3
P_5	0.1	0.3	0.6

(CD)

	C_1	C_2	C_3
P_1	0	0.6	0
P_2	0	0	0
P_3	0.4	0	0
P_4	0	0	0
P_5	0	0	0.2

(PCD)

Figura 3.6: Comparação de CD com o PCD

A disposição das premissas para as classes, de acordo com o CD, seria da forma $C_1 : \{P_2, P_3, P_4\}$, $C_2 : \{P_1, P_4\}$ e $C_3 : \{P_2, P_5\}$; para o PCD, da forma $C_1 : \{P_3\}$, $C_2 : \{P_1\}$ e $C_3 : \{P_5\}$.

3.2.3.3 Frequência Máxima

É uma variação do CD, binarizando-se os graus de pertinência. Neste caso, as premissas são associadas à classe que maximiza a equação:

$$AssociaçãoFreq_k = \frac{\sum_{i \in k} I_i^{\{0,1\}}}{\sum_{i=1}^n I_i^{\{0,1\}}}, I_i^{\{0,1\}} = \begin{cases} 1 & , \text{ se } \mu_{A_d}(\mathbf{x}_i) > 0 \\ 0 & , \text{ se } \mu_{A_d}(\mathbf{x}_i) = 0 \end{cases} \quad (3-7)$$

3.2.3.4 Mínimos Quadrados Restritos

Este método parte do princípio de que toda premissa pode ser reescrita como uma combinação convexa das K classes que busca explicar. Nesse sentido, para cada d -ésima premissa, determina-se uma configuração linear de pesos, no intervalo $[0, 1]$, o que pode ser descrito como um problema de otimização:

$$\begin{aligned} \min \quad & \sum_{i=1}^n (\mu_{A_d}(\mathbf{x}_i) - \sum_{k=1}^K \beta_k \mu_{C_i \in k}(\mathbf{x}_i))^2 \\ \text{sujeito a} \quad & \sum_{k=1}^K \beta_k = 1 \text{ e } \beta_k \geq 0 \end{aligned} \quad (3-8)$$

onde $\mu_{C_i \in k}(\mathbf{x}_i) \in \{0, 1\}$ é o grau de pertinência observado do padrão \mathbf{x}_i à classe k e β_k é o grau de influência da classe k na descrição da premissa $\mu_{A_d}(\mathbf{x}_i)$. Este é um problema típico de Programação Quadrática, cuja solução é a mesma de [18] com o uso dos Gradientes Conjugados. A classe k descarta premissas $\mu_{A_d}(\mathbf{x}_i)$ com $\beta_k = 0$. Se $\beta_k = 0$ para todo k , então a premissa não é endereçada a nenhum termo consequente.

Este problema pode ser descrito, para cada classe k , por:

$$\begin{aligned} \min \quad & \|C_k - U^{\hat{P}} * B_k\| \\ \text{sujeito a} \quad & \sum_{p=1}^P \beta_k^p = 1 \text{ e } \beta_k^p \geq 0 \end{aligned} \quad (3-9)$$

onde $U^{\hat{P}}$ representa os graus de pertinência de cada um dos n padrões relativos a cada premissa p . Procura-se ajustar um vetor de pesos B_k que associam $U^{\hat{P}}$ com C_k (representação binária da classe k).

$$U^{\hat{P}} = (U^1 \mid \dots \mid U^p \mid \dots \mid U^P), \quad B = (B_1 \mid \dots \mid B_k \mid \dots \mid B_K)$$

$$U^{\hat{P}} = \left(\begin{array}{c|c|c|c|c} \mu_1^1 & \dots & \mu_1^p & \dots & \mu_1^P \\ \vdots & & \vdots & & \vdots \\ \mu_n^1 & \dots & \mu_n^p & \dots & \mu_n^P \end{array} \right), \quad B = \left(\begin{array}{c|c|c|c|c} \beta_1^1 & \dots & \beta_k^1 & \dots & \beta_K^1 \\ \vdots & & \vdots & & \vdots \\ \beta_1^p & \dots & \beta_k^p & \dots & \beta_K^p \\ \vdots & & \vdots & & \vdots \\ \beta_1^P & \dots & \beta_k^P & \dots & \beta_K^P \end{array} \right)$$

$$C = (C_1 \mid \dots \mid C_k \mid \dots \mid C_K)$$

$$C = \left(\begin{array}{c|c|c|c|c} c_1^1 & \dots & c_k^1 & \dots & c_K^1 \\ \vdots & & \vdots & & \vdots \\ c_1^n & \dots & c_k^n & \dots & c_K^n \end{array} \right)$$

As premissas associadas a uma classe k são aquelas com $\beta_k^p \neq 0$. Neste tipo de associação, é permitido que uma premissa possa ser endereçada a vários consequentes; isto pode ser de utilidade para algumas bases de múltiplas classes que intrinsecamente compartilhem certas características. Cumpre salientar que isto pode levar a problemas de interpretabilidade.

3.2.4 Agregação

Em um Sistema de Inferência Fuzzy, um novo padrão \mathbf{x}_i^* pode ser compatível com diversas regras pertinentes a uma ou mais classes. A etapa de agregação neste modelo tem como objetivo juntar os graus de ativação de \mathbf{x}_i^* para as diferentes regras de uma mesma classe, de maneira a gerar um valor consensual para a discriminação da classe alvo.

Considere-se $D^{(k)}$ o número de regras fuzzy associadas à k -ésima classe. Dado um operador de agregação [42, 43], $g : [0, 1]^{D^{(k)}} \rightarrow [0, 1]$, o grau de pertinência de \mathbf{x}_i^* a cada uma das K classes ($\hat{\mu}_{C_i \in k}(\mathbf{x}_i^*)$) é:

$$\hat{\mu}_{C_i \in 1}(\mathbf{x}_i^*) = g[\mu_{A_{1(1)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(1)}}(\mathbf{x}_i^*)] \quad (3-10)$$

$$\hat{\mu}_{C_i \in 2}(\mathbf{x}_i^*) = g[\mu_{A_{1(2)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(2)}}(\mathbf{x}_i^*)] \quad (3-11)$$

...

$$\hat{\mu}_{C_i \in K}(\mathbf{x}_i^*) = g[\mu_{A_{1(K)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(K)}}(\mathbf{x}_i^*)] \quad (3-12)$$

A seguir são descritos os operadores de agregação aqui utilizados.

3.2.4.1 Máximo

A t-conorma mais usada como operador de agregação em Sistemas de Inferência Fuzzy é o máximo, ilustrado na Figura 3.7. As premissas relacionadas à classe C_1 são P_d , P_h e P_v ; para a classe C_2 , (P_g , P_r e P_w).

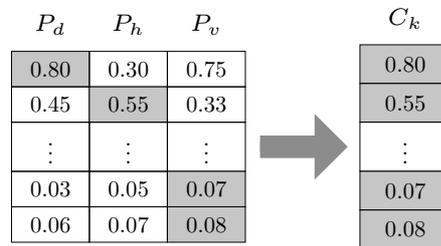


Figura 3.7: Agregação usando o máximo.

3.2.4.2 MQR

Uma alternativa é o uso de uma média ponderada, similarmente ao procedimento do MQR descrito em 3.2.3.4, na etapa de Associação. Propõe-se, assim, o uso da Média Ponderada estimada via Mínimos Quadrados Restritos (MPMQR) por classe:

$$\hat{\mu}_{C_i \in k}(\mathbf{x}_i^*) = \sum_{d^{(1)}=1}^{D^{(k)}} w_{d^{(k)}} \mu_{A_{d^{(k)}}}(\mathbf{x}_i) \quad (3-13)$$

onde $w_{d^{(k)}}$ é o peso ou grau de influência de $\mu_{A_{d^{(k)}}}(\mathbf{x}_i)$ na discriminação de padrões oriundos da classe k , o que equivale à equação 3-8 do MQR.

3.2.4.3 IntMQR

Concluída a etapa da Associação, as premissas sobreviventes para cada classe são combinadas entre elas com o uso de uma t-norma (por exemplo, o produto). A Figura 3.8 ilustra o processo descrito, onde as premissas sobreviventes são P_d , P_h e P_v . Em consequência, P_{dh} , P_{dv} e P_{hv} representam as combinações.

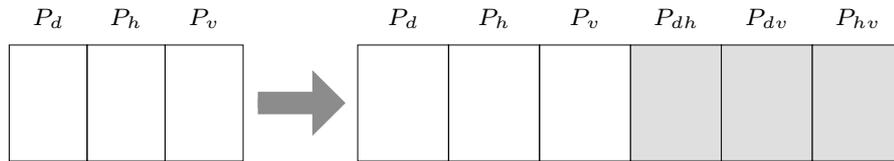


Figura 3.8: Inter-relação das premissas sobreviventes da Associação

Finalmente, aplica-se o MQR para o cálculo dos pesos do novo universo de premissas por classe. As premissas com peso zero ($\beta_k = 0$) são desconsideradas e as restantes compõem a base de regras da respectiva classe.

Computado o grau de pertinência de \mathbf{x}_i^* a cada uma das K classes, verifica-se, na etapa de Decisão, a que classe \mathbf{x}_i^* pertence pelas informações dadas por $\hat{\mu}_{C_i \in 1}(\mathbf{x}_i^*), \dots, \hat{\mu}_{C_i \in K}(\mathbf{x}_i^*)$.

3.2.5 Decisão

Dado um novo padrão \mathbf{x}_i^* , a decisão sobre a sua pertinência à k -ésima classe é realizada por:

$$\hat{C}_i = \arg_k \max \{ \hat{\mu}_{C_i \in 1}(\mathbf{x}_i^*), \dots, \hat{\mu}_{C_i \in k}(\mathbf{x}_i^*), \dots, \hat{\mu}_{C_i \in K}(\mathbf{x}_i^*) \} \quad (3-14)$$

onde \hat{C}_i é a classe predita, resultado do k -ésimo argumento que maximiza a expressão (3-14). Assim, esse método associa o padrão \mathbf{x}_i^* à classe à qual ele é mais pertinente, de acordo com a base de regras disponível. Quando há

empates, uma heurística decisória pode ser aplicada (por exemplo, impor a \mathbf{x}_i^* a classe que possui mais padrões na base de dados) ou nenhuma classe é definida para \mathbf{x}_i^* .

Com as informações de \hat{C}_i para todo o padrão, é possível avaliar a qualidade do modelo AutoFIS. Uma forma simples é o Erro Médio de Classificação (*EMC*):

$$EMC = \frac{\sum_{i=1}^n |C_i - \hat{C}_i|}{n} \quad (3-15)$$

onde $|C_i - \hat{C}_i| = 0$ se $C_i = \hat{C}_i$, e 1 caso contrário.

3.3 Resumo

Apresentou-se, neste capítulo, o AutoFIS-Class, um modelo de Síntese Automática de Sistemas de Inferência Fuzzy para Classificação. A ideia central é gerar premissas que garantam critérios mínimos de qualidade para reduzir assim o espaço de busca e associar cada premissa a um termo consequente mais compatível.

O próximo capítulo apresenta os testes realizados com modelo AutoFIS-Class. Os resultados são comparados com os obtidos com outros modelos.

4

Estudo de Casos

Este capítulo apresenta uma série de estudos de casos, cuja finalidade principal é exibir a qualidade das soluções fornecidas pelo modelo. A seguir, o formato adotado:

- **Primeiro tópico - Investigação empírica:** determina-se a configuração mais competitiva para o modelo AutoFIS-Class, visando à futura comparação com os demais modelos (SFEs ou outros). São apresentadas as bases de dados benchmarks, a justificativa da escolha (pela preferência de SFEs) e o procedimento experimental realizado. Em seguida, exibem-se as análises que determinam a configuração *default* a ser usada pelo AutoFIS-Class.
- **Segundo tópico - Comparação com outros modelos:** a partir da configuração encontrada para o modelo AutoFIS-Class, comparam-se os seus resultados com os dos demais métodos (alguns SFE e outros modelos). Avalia-se tanto a acurácia quanto a complexidade do sistema (número de regras geradas).

Segue-se a abordagem (descrita em [37]) de efetuar análises estatísticas sempre que possível, com o objetivo de verificar qual método ou configuração apresenta resultados significativamente superiores, dado um certo conjunto de bases de dados. Para verificar se a diferença da acurácia (procura-se a maximização) e do número de regras (procura-se a minimização) entre os modelos é significativa, são aplicados os testes de Friedman, Iman-Davenport e pós-teste de Holm [44]. Utilizam-se as rotinas estatísticas do KEEL [45] e o pacote estatístico R. As bases de dados usadas nas aplicações benchmark foram em sua maioria obtidas da University of California Irvine (UCI) [46] e do repositório KEEL [45].

Todas as rotinas executadas foram implementadas em Python 2.7.9 e executadas principalmente em um PC Windows 7 com processador Intel i7 3.4GHz, 16GB de RAM. Alguns experimentos iniciais foram realizados em dois PCs com processadores Intel i7 3.2Ghz, 32GB de RAM, com Windows e Linux; o que permitiu entender o alcance do espaço gerado pelo modelo proposto e os ajustes correspondentes visando à escalabilidade do modelo em computadores com menor memória RAM.

Foram consideradas 48 bases de dados, agrupadas em bases binárias e de múltiplas classes, cujas principais informações são mostradas nas Tabelas 4.1 e 4.2 respectivamente.

	Base de dados	Classes	Atributos	Instâncias
1	appendicitis	2	7	106
2	australian	2	14	690
3	banana	2	2	5300
4	bupa	2	6	345
5	crx	2	15	653
6	coil2000	2	85	9822
7	german	2	20	1000
8	haberman	2	3	306
9	heart	2	13	210
10	hepatitis	2	19	155
11	ionosphere	2	33	251
12	magic	2	10	19020
13	monk-2	2	6	432
14	phoneme	2	5	5404
15	pima	2	8	768
16	ring	2	20	7400
17	saheart	2	9	462
18	sonar	2	60	208
19	spambase	2	57	4597
20	spectfheart	2	44	267
21	titanic	2	3	2201
22	two-norm	2	20	7400
23	wdbc	2	30	569
24	wisconsin	2	9	683

Tabela 4.1: Principais características das bases de dados binárias.

	Base de dados	Classes	Atributos	Instâncias
1	automobile	6	25	205
2	balance	3	4	625
3	contraceptive	3	9	1473
4	cleveland	5	13	297
5	dermatology	6	34	358
6	ecoli	8	7	336
7	glass	6	9	214
8	hayes-Roth	3	4	160
9	iris	3	4	150
10	movementlibras	15	90	360
11	new-thyroid	3	5	215
12	optdigits	10	64	5620
13	page-blocks	5	10	5472
14	penbased	10	16	10992
15	satimage	6	36	6435
16	segment	7	19	2310
17	tae	3	5	151
18	texture	11	40	5500
19	thyroid	3	21	7200
20	vehicle	4	18	846
21	vowel	11	13	990
22	wine	3	13	178
23	winequality-Red	11	11	1599
24	yeast	10	8	1484

Tabela 4.2: Principais características das bases de dados de múltiplas classes.

As 48 bases de dados foram escolhidas para permitir uma comparação com outros modelos da literatura apresentados na Seção 2.1, como também para avaliar o modelo frente a um número elevado de atributos. As rotinas do modelo AutoFIS-Class seguiram as linhas de Alcalá et al. [35]:

- Validação cruzada de 10 pastas (10-fold-cv).
- Obtenção da acurácia do teste.
- Obtenção do número de regras.

4.1 Investigação Empírica da Arquitetura do Modelo AutoFIS-Class

Nesta investigação, busca-se identificar, por meio de um experimento sequencial, a configuração geral mais apropriada em termos de acurácia e do número de regras para o modelo AutoFIS-Class. Por meio de testes preliminares, definiu-se o delineamento sequencial apresentado na Figura 4.1. Este procedimento proporcionou a geração de resultados para a maioria das bases de dados. Note-se que a partir de agora o termo Cardinalidade faz referência ao Filtro de Suporte Relativo.

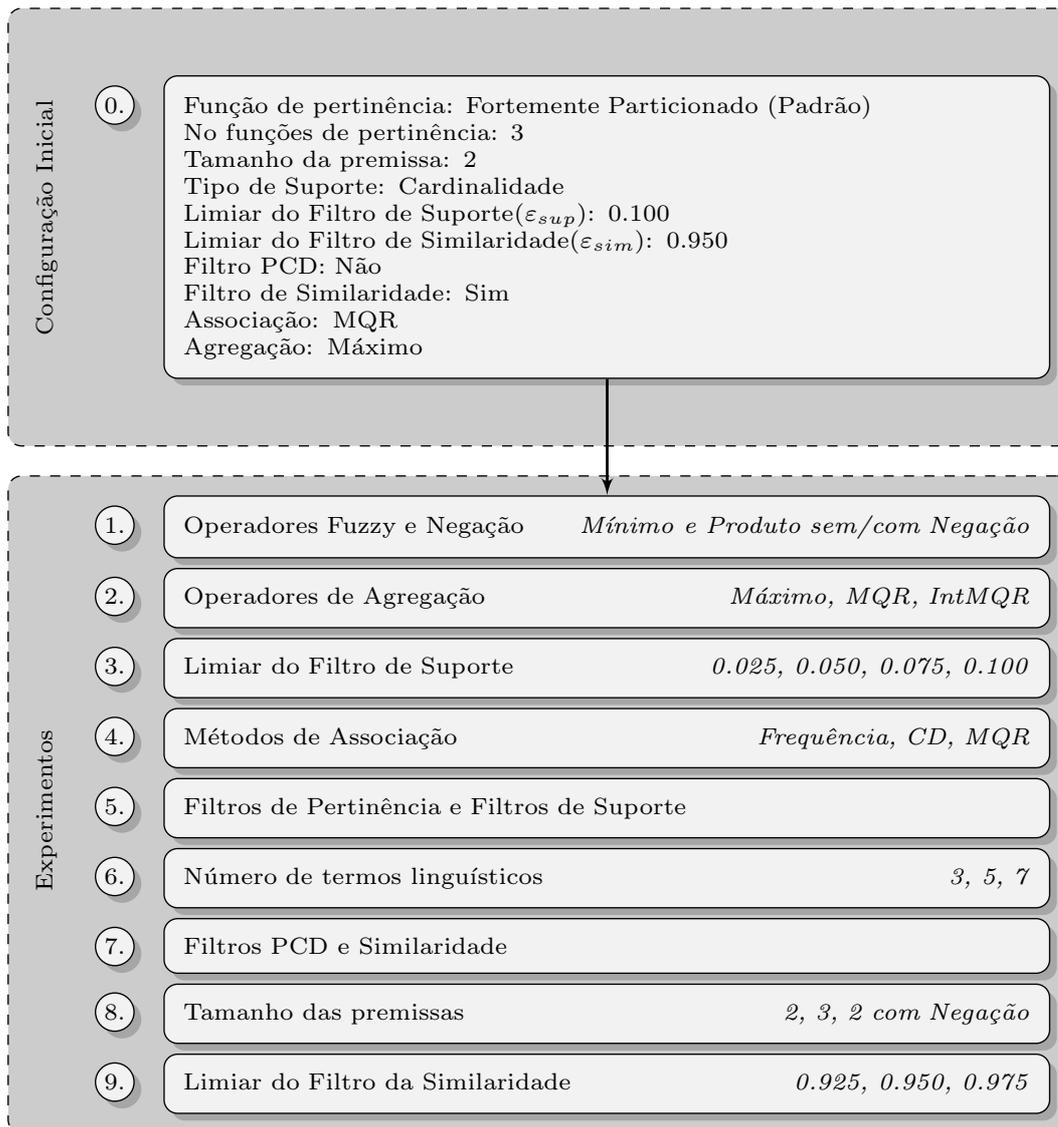


Figura 4.1: Delineamento sequencial dos experimentos com o modelo AutoFIS-Class

Cada um dos nove experimentos realizados será descrito em detalhe a seguir. Em função do elevado custo computacional requerido em algumas bases de dados – devido à quantidade considerável de premissas geradas na etapa de Formulação –, a investigação empírica da arquitetura do modelo AutoFIS-Class considerou um conjunto reduzido de bases de dados, mostradas na Tabela 4.3. Posteriormente, para a etapa de comparação de AutoFIS-Class com outros modelos, considera-se um conjunto maior de bases de dados.

Base de dados	
Binárias	Múltiplas classes
appendicitis	automobile
australian	balance
banana	contraceptive
bupa	cleveland
crx	ecoli
haberman	glass
heart	hayes-roth
hepatitis	iris
magic	new-thyroid
monk-2	tae
phoneme	thyroid
pima	vehicle
saheart	vowel
titanic	wine
wisconsin	winequality-red
	yeast

Tabela 4.3: Bases de dados utilizadas para gerar a arquitetura empírica do Modelo AutoFIS-Class.

4.1.1 Experimento 1 - Avaliação dos Operadores Fuzzy

O experimento 1 destinou-se a investigar a influência das diferentes configurações do conjunto de Operadores Fuzzy. A hipótese central é avaliar o efeito dos operadores Mínimo e Produto na acurácia e no número de regras. Considerou-se também o uso do operador de Negação. A configuração considerada é apresentada na Figura 4.2.



Figura 4.2: Experimento de avaliação da t-norma e operador de negação

Resultados

As acurácias e o número de regras das configurações do Experimento 1 (Tabela 4.2) são apresentadas na Tabela 4.4. Os melhores resultados em termos de acurácia aparecem em negrito e destaca-se, em fundo cinza, o menor número de regras para cada base de dados. Os resultados do teste de Friedman e Holm para a acurácia e para o número de regras são apresentados na Tabela 4.5 e na Tabela 4.6, respectivamente.

Base de dados	Acurácia				Número de regras				
	Mínimo		Produto		Mínimo		Produto		
	-	Neg	-	Neg	-	Neg	-	Neg	
1	appendicitis	65.18	79.27	65.18	79.27	10.40	10.50	9.70	10.50
2	australian	78.41	55.51	79.57	55.51	20.20	27.90	20.50	28.00
3	balance	62.10	52.95	62.10	52.31	57.20	50.30	57.30	54.40
4	banana	56.04	59.66	56.04	59.64	12.80	14.60	10.80	15.40
5	bupa	55.54	57.60	56.11	57.88	15.80	17.90	14.30	14.70
6	cleveland	56.20	55.22	56.54	54.54	76.20	76.60	75.00	76.90
7	contraceptive	43.25	44.40	43.59	43.18	45.20	57.10	49.00	55.60
8	crx	72.04	68.63	74.60	73.77	20.30	24.70	20.60	24.50
9	ecoli	35.72	49.12	36.34	50.59	84.40	74.90	76.80	76.90
10	glass	43.52	41.91	45.51	43.67	40.10	41.10	37.00	37.50
11	haberman	69.28	72.52	69.28	73.52	16.50	29.40	16.30	29.20
12	hayes-roth	45.00	40.62	45.00	40.62	33.50	37.10	33.50	37.00
13	heart	67.41	60.00	68.15	58.52	22.40	27.30	21.10	27.80
14	hepatitis	83.43	83.43	83.43	83.43	25.50	26.80	26.00	27.50
15	iris	76.67	64.67	76.67	70.67	12.10	14.90	11.90	15.30
16	magic	64.29	75.22	65.04	73.31	13.40	11.20	12.70	10.10
17	monk-2	52.66	58.58	52.66	58.58	30.10	24.20	30.10	24.20
18	newthyroid	77.29	79.16	77.29	79.63	14.90	14.60	12.50	14.10
19	page-blocks	86.90	88.60	86.90	89.78	26.90	26.70	28.40	27.20
20	phoneme	59.49	74.72	59.36	74.78	16.80	20.00	14.80	17.80
21	pima	58.07	66.65	56.51	67.72	16.20	17.70	15.10	15.80
22	saheart	60.59	67.97	58.44	68.40	21.60	22.20	19.00	21.80
23	tae	34.42	34.42	34.42	34.42	32.70	40.20	32.70	40.20
24	thyroid	92.54	92.58	92.58	92.58	19.90	17.70	17.80	17.40
25	titanic	67.74	67.74	67.74	67.74	14.20	23.30	14.20	23.30
26	vehicle	43.03	37.60	42.91	36.04	33.40	34.10	30.50	30.20
27	vowel	18.38	15.66	21.72	17.68	142.70	112.80	130.50	106.30
28	wine	72.03	83.79	72.06	82.09	17.60	20.90	17.60	21.40
29	winequality-red	48.53	46.15	48.09	46.34	48.70	51.60	46.90	47.20
30	wisconsin	89.31	80.95	90.78	79.50	21.70	22.10	20.60	22.00
31	yeast		46.22	39.35	46.22		59.50	54.90	58.70

Tabela 4.4: Acurácias e número de regras no Experimento 1.

i	Configuração	Posto
3	Min	2.6562
2	MinNeg	2.5469
1	ProdNeg	2.4219
0	Prod	2.3750

Teste	p-valor
Friedman	<0.8186
Iman e Davepor	<0.8234

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
Min	0.8714	0.3835	0.0167	Não
MinNeg	0.5325	0.5944	0.0250	Não
ProdNeg	0.1452	0.8845	0.0500	Não

Tabela 4.5: Resultados do teste de Friedman e Holm para a acurácia do Experimento 1.

i	Configuração	Posto
3	MinNeg	3.0312
2	ProdNeg	2.7812
1	Min	2.4219
0	Prod	1.7656

Teste	p-valor
Friedman	<0.0005
Iman e Davepor	<0.0003

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
MinNeg	3.9214	0.0001	0.0167	Sim
ProdNeg	3.1468	0.0017	0.0250	Sim
Min	2.0333	0.0420	0.0500	Sim

Tabela 4.6: Resultados do teste de Friedman e Holm para o número de regras do Experimento 1.

Discussões

- No teste de Friedman e Holm para a acurácia (%), a configuração Produto apresentou menor posto, mas esta não é uma configuração substancialmente superior às outras do Experimento 1. Na minimização do número de regras, atestado pelo teste de Friedman e Holm, a configuração Produto foi superior às outras configurações, confirmando-se a capacidade de contração desta t-norma. Baseado nesses resultados, o operador fuzzy escolhido foi o Produto.
- Os resultados indicam que o uso de outros operadores fuzzy, fora do âmbito das t-normas, como o operador de Negação, possibilita ganhos de acurácia em problemas de classificação.

É interessante observar e comentar o comportamento da Negação: sua adição na t-norma Produto possibilitou que 15 bases de dados obtivessem os melhores resultados. Entretanto, seu uso acarreta um aumento no tempo computacional de ajuste do modelo. A Tabela 4.7 mostra o fator de incremento no tempo computacional do uso do operador de Negação. Este fator é calculado dividindo-se o tempo observado com a configuração com a Negação pelo o tempo da configuração sem a Negação. Apesar do aumento no custo computacional, a acurácia foi melhorada para várias bases de dados. Por esta razão, não se deve descartar totalmente o uso do operador de Negação.

	Base de dados	Fator da Negação	
		Mínimo	Produto
1	appendicitis	2.38	3.78
2	australian	14.64	15.85
3	automobile	22.19	28.53
4	balance	3.58	3.64
5	banana	1.27	1.17
6	bupa	3.08	3.17
7	cleveland	12.63	13.64
8	contraceptive	19.26	16.76
9	crx	8.60	7.45
10	ecoli	5.04	4.44
11	glass	3.84	3.54
12	haberman	2.58	2.40
13	hayes-roth	5.31	5.55
14	heart	14.09	13.70
15	hepatitis	3.58	3.49
16	iris	3.21	3.14
17	magic	12.76	17.80
18	monk-2	4.54	5.42
19	newthyroid	2.66	3.84
20	page-blocks	1.97	2.45
21	phoneme	10.06	12.01
22	pima	8.78	9.30
23	saheart	8.06	8.31
24	tae	8.51	8.35
25	thyroid	4.33	4.07
26	titanic	1.50	1.40
27	vehicle	27.85	39.03
28	vowel	65.59	82.57
29	wine	20.13	30.33
30	winequality-red	14.09	14.79
31	wisconsin	29.14	26.61

Tabela 4.7: Fator de incremento em tempo por uso do operador de Negação nas configurações de Mínimo e Produto.

4.1.2 Experimento 2 - Operadores de Agregação

No experimento, investigou-se a influência dos Operadores de Agregação Máximo, MQR e IntMQR (Seção 3.2). Partindo da configuração encontrada no Experimento 1, a Figura 4.3 apresenta as configurações agora avaliadas.

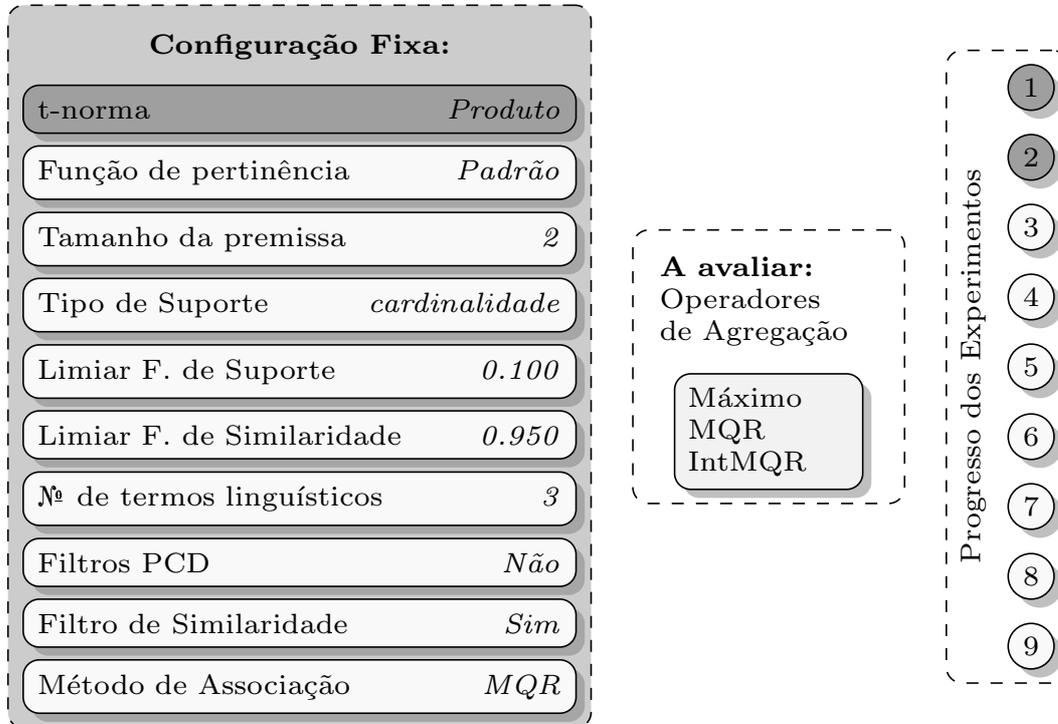


Figura 4.3: Experimento de avaliação dos Operadores de Agregação.

Resultados

As acurácias e o número de regras das configurações do Experimento 2 são apresentadas na Tabela 4.8. Novamente, em negrito são apresentados os melhores resultados em termos de acurácia, assim como, em fundo cinza, o menor número de regras para cada base de dados. As análises de significância pelo teste de Friedman e Holm, para a acurácia e o número de regras, são apresentadas nas Tabelas 4.9 e 4.10, respectivamente.

	Base de dados	Acurácia			Número de regras		
		Máximo	MQR	IntMQR	Máximo	MQR	IntMQR
1	appendicitis	65.18	87.09	87.91	9.70	5.30	6.70
2	australian	79.57	85.51	85.80	20.50	13.80	18.40
3	balance	62.10	89.59	78.59	57.30	23.80	23.80
4	banana	56.04	60.21	52.64	10.80	6.00	4.20
5	bupa	56.11	62.33	58.81	14.30	7.50	8.30
6	cleveland	56.54	59.27	58.63	75.00	48.40	52.70
7	contraceptive	43.59	52.07	50.58	49.00	28.70	28.40
8	crx	74.60	86.30	86.45	20.60	13.10	15.00
9	ecoli	36.34	72.89	71.12	76.80	46.60	73.50
10	glass	45.51	58.68	50.81	37.00	21.00	27.60
11	haberman	69.28	75.13	73.51	16.30	9.10	8.70
12	hayes-roth	45.00	64.38	56.25	33.50	17.90	78.20
13	heart	68.15	86.30	84.81	21.10	13.10	17.20
14	hepatitis	83.43	85.23	83.65	26.00	15.50	14.80
15	iris	76.67	95.33	96.67	11.90	6.00	6.80
16	magic	65.04	77.38	76.37	12.70	6.50	5.90
17	monk-2	52.66	91.10	94.09	30.10	9.70	6.80
18	newthyroid	77.29	84.74	83.79	12.50	8.40	10.10
19	page-blocks	86.90	89.80	89.80	28.40	15.70	16.30
20	phoneme	59.36	70.54	70.67	14.80	7.90	8.10
21	pima	56.51	75.27	73.97	15.10	8.70	11.00
22	saheart	58.44	73.83	73.39	19.00	9.60	12.20
23	tae	34.42	47.17	43.12	32.70	17.60	29.30
24	thyroid	92.58	92.21	92.58	17.80	10.60	23.30
25	titanic	67.74	77.65	78.28	14.20	9.70	13.80
26	vehicle	42.91	56.16	58.15	30.50	18.00	19.60
27	vowel	21.72	45.66	45.45	130.50	69.80	65.70
28	wine	72.06	91.47	86.93	17.60	9.60	10.80
29	winequality-red	48.09	57.85	57.66	46.90	27.20	33.60
30	wisconsin	90.78	94.45	96.08	20.60	13.70	12.50
31	yeast	39.35	51.62	51.01	54.90	27.60	37.10

Tabela 4.8: Acurácias das bases de dados no Experimento 1.

i	Configuração	Posto
2	Maximo	2.9194
1	IntMQR	1.7097
0	MQR	1.3710

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
Maximo	6.0960	0.0001	0.0250	Sim
IntMQR	1.3335	0.1824	0.0500	Não

Tabela 4.9: Teste de Friedman e Holm para a acurácia.

i	Configuração	Posto
2	Maximo	2.9394
1	IntMQR	1.7727
0	MQR	1.2879

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
Maximo	6.7085	0.0001	0.0250	Sim
IntMQR	1.9695	0.0489	0.0500	Sim

Tabela 4.10: Teste de Friedman e Holm para o número de regras.

Discussões

No teste de Friedman e Holm para a acurácia, a configuração MQR obtém um melhor ranking (menor posto) do que IntMQR, mas não é substancialmente superior, conforme mostrado na Tabela 4.9. Por outro lado, na minimização do número de regras, o MQR é substancialmente superior (Tabela 4.10) ao IntMQR. Com base nestes resultados, o MQR foi escolhido como o operador de agregação para os experimentos seguintes.

4.1.3 Experimento 3 - Limiar do Filtro do Suporte

Investiga-se, neste experimento, um valor para o Limiar do Filtro de Suporte. Quatro valores são avaliados: 0.025, 0.050, 0.075 e 0.100, ou seja, são consideradas premissas que cobrem pelo menos 2.5%, 5%, 7.5% e 10% de instâncias nas bases de dados. A partir da configuração definida no Experimento 2, a Figura 4.4 apresenta as configurações avaliadas no presente experimento.

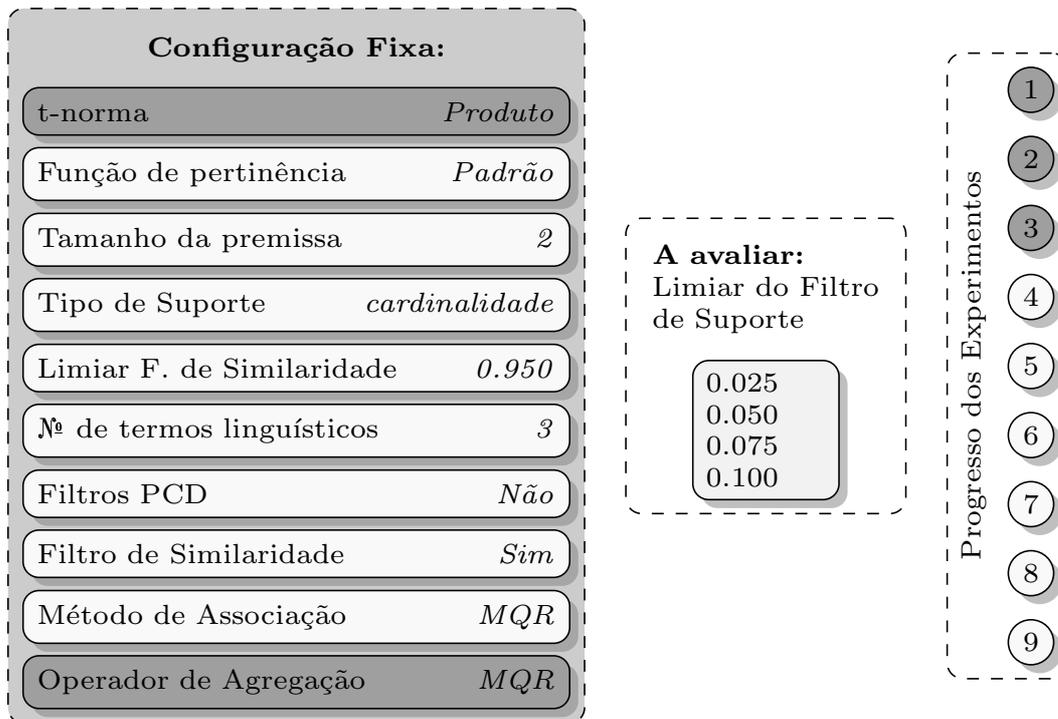


Figura 4.4: Experimento de avaliação do limiar do Filtro de Suporte.

Resultados

Os resultados obtidos neste experimento são apresentados na Tabela 4.11. O resultado do teste de Friedman e Holm das acurácias é apresentado na Tabela 4.12; o referente ao número de regras, na Tabela 4.13.

Base de dados	Acurácia				Número de regras				
	0.025	0.050	0.075	0.100	0.025	0.050	0.075	0.100	
1	appendicitis	87.09	86.18	87.00	87.09	5.90	5.60	5.50	5.30
2	australian	85.65	85.51	85.65	85.51	13.10	14.40	11.60	13.80
3	balance	89.76	89.76	89.76	89.59	24.90	24.90	24.90	23.80
4	banana	56.60	63.00	63.00	60.21	7.70	5.80	5.80	6.00
5	bupa	60.54	60.29	62.62	62.33	7.70	8.20	8.00	7.50
6	cleveland	57.95	59.63	58.96	59.27	55.80	53.60	50.30	48.40
7	contraceptive	51.67	51.40	52.55	52.07	30.50	30.40	29.60	28.70
8	crx	86.15	86.30	86.30	86.30	12.10	12.40	11.90	13.10
9	ecoli	72.61	73.20	73.79	72.89	48.20	44.90	48.90	46.60
10	glass	60.59	60.86	60.50	58.68	26.60	24.40	24.40	21.00
11	haberman	74.48	74.81	75.14	75.13	10.00	9.90	9.30	9.10
12	hayes-roth	70.00	70.62	70.62	64.38	11.50	11.20	11.20	17.90
13	heart	84.81	84.44	84.44	86.30	14.10	14.00	13.70	13.10
14	hepatitis	83.42	83.42	85.23	85.23	16.20	16.00	16.80	15.50
15	iris	93.33	95.33	95.33	95.33	6.50	6.10	6.00	6.00
16	magic	76.68	77.38	77.21	77.38	5.20	6.50	6.60	6.50
17	monk-2	92.24	92.24	92.24	91.10	9.70	9.70	9.60	9.70
18	newthyroid	85.17	86.56	86.58	84.74	8.60	8.80	8.60	8.40
19	page-blocks	81.61	88.89	89.80	89.80	16.10	15.50	15.70	15.70
20	phoneme	70.34	70.26	70.37	70.54	8.20	8.10	7.90	7.90
21	pima	75.53	75.01	74.50	75.27	8.40	7.90	8.70	8.70
22	saheart	72.52	71.45	72.30	73.83	11.00	10.40	11.00	9.60
23	tae	49.04	45.62	43.12	47.17	22.20	20.00	17.80	17.60
24	thyroid	92.40	92.54	92.31	92.21	17.80	11.80	10.00	10.60
25	titanic	77.51	77.60	77.60	77.65	9.70	9.50	8.80	9.70
26	vehicle	57.10	55.33	56.27	56.16	19.30	19.20	18.00	18.00
27	vowel	47.98	45.05	45.15	45.66	78.90	84.20	67.10	69.80
28	wine	93.73	95.46	92.65	91.47	10.60	10.20	10.60	9.60
29	winequality-red	57.41	57.04	57.29	57.85	31.60	31.50	28.10	27.20
30	wisconsin	94.60	94.91	94.90	94.45	13.40	13.00	13.80	13.70
31	yeast	54.18	52.29	50.40	51.62	36.80	38.60	31.60	27.60

Tabela 4.11: Acurácia e número de regras no Experimento 3.

i	Configuração	Posto
3	0.025	2.6290
2	0.050	2.5968
1	0.100	2.4677
0	0.075	2.3065

Teste	p-valor
Friedman	<0.7530
Iman e Davepor	<0.7589

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
3	0.9837	0.2526	0.0167	Não
2	0.8853	0.3404	0.0250	Não
1	0.4919	0.5673	0.0500	Não

Tabela 4.12: Resultados do teste de Friedman e Holm para a acurácia do Experimento 3.

i	Configuração	Posto
3	0.025	3.2581
2	0.050	2.6452
1	0.075	2.2419
0	0.100	1.8548

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
3	4.2792	0.0001	0.0167	Sim
2	2.4102	0.0159	0.0250	Sim
1	1.1805	0.2378	0.0500	Não

Tabela 4.13: Resultados do teste de Friedman e Holm para o número de regras do Experimento 3.

Discussões

Como esperado, para o limiar de 0.100 o número de regras gerado foi menor. Em termos de acurácia, o limiar 0.075 consegue um melhor ranking (menor posto), mas não é substancialmente superior aos outros de acordo com o teste de Friedman e Holm (Tabela 4.12). Já para o número de regras, os valores de limiar 0.075 e 0.100 são substancialmente superiores aos outros (0.050 e 0.025), conforme mostrado na (Tabela 4.13). Escolheu-se, assim, a configuração com valor 0.075.

4.1.4 Experimento 4 - Métodos de Associação

O experimento 4 avaliou a influência dos Métodos de Associação Frequência Máxima, CD e MQR. A partir da configuração encontrada no Experimento 3, a Figura 4.5 apresenta as configurações deste experimento.

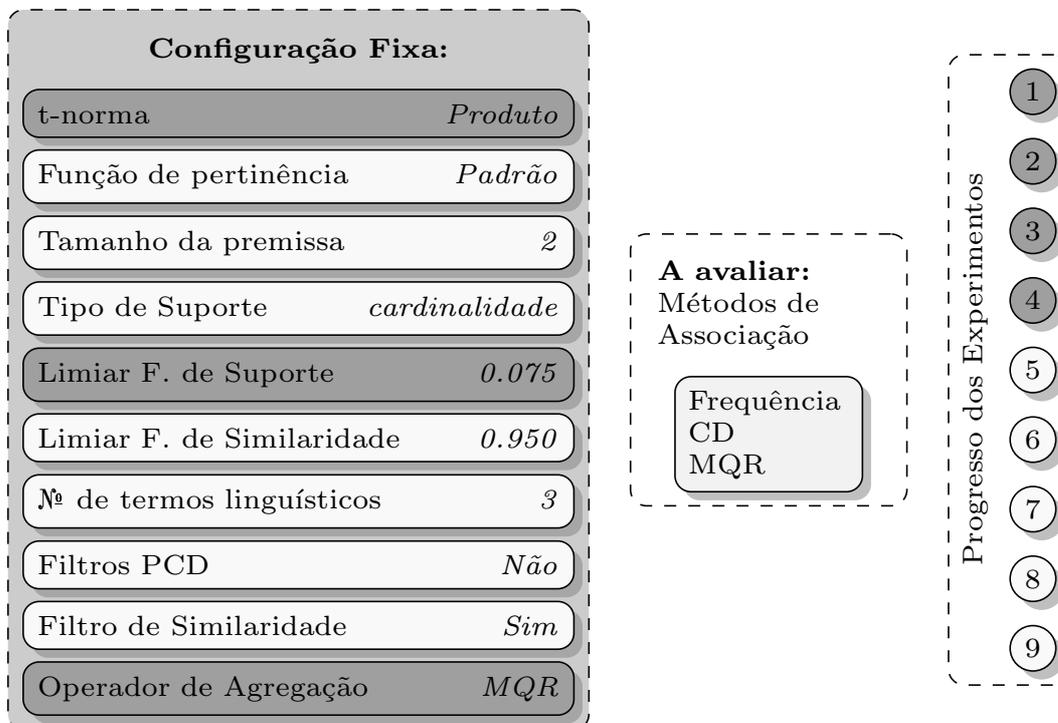


Figura 4.5: Experimento de avaliação dos Métodos de Associação

Resultados

As acurácias e o número de regras das configurações do Experimento 4 são apresentadas na Tabela 4.14.

	Base de dados	Acurácia			Número de regras		
		Freq	CD	MQR	Freq	CD	MQR
1	appendicitis		87.91	87.00		4.70	5.50
2	australian	85.51	85.51	85.65	14.10	14.10	11.60
3	banana	56.15	58.75	63.00	3.70	3.00	5.80
4	bupa			62.62			8.00
5	crx	86.30	86.30	86.30	14.40	14.00	11.90
6	haberman			75.14			9.30
7	heart	83.33	83.33	84.44	14.30	13.90	13.70
8	hepatitis	83.42	83.42	85.23	9.60	9.10	16.80
9	magic	71.62	74.00	77.21	4.30	4.40	6.60
10	monk-2	96.81	96.81	92.24	7.40	7.40	9.60
11	phoneme	71.30	71.30	70.37	3.30	3.30	7.90
12	pima		75.67	74.50		6.00	8.70
13	saheart	69.70	72.52	72.30	7.80	7.20	11.00
14	titanic	77.60	77.60	77.60	10.60	10.60	8.80
15	wisconsin	94.47	94.16	94.90	10.60	11.30	13.80
1	balance			89.76			24.90
2	cleveland			58.96			50.30
3	contraceptive	50.44	52.62	52.55	22.90	23.30	29.60
4	ecoli			73.79			48.90
5	glass			60.50			24.40
6	hayes-roth	68.75	68.75	70.62	10.80	10.80	11.20
7	iris	96.00	96.00	95.33	3.90	3.90	6.00
8	newthyroid			86.58			8.60
9	page-blocks			89.80			15.70
10	tae	43.12	47.67	43.12	14.90	15.50	17.80
11	thyroid			92.31			10.00
12	vehicle	55.92	55.32	56.27	15.60	18.30	18.00
13	vowel		38.08	45.15		59.20	67.10
14	wine	87.71	93.17	92.65	7.90	9.40	10.60
15	winequality-red			57.29			28.10
16	yeast			50.40			31.60

Tabela 4.14: Acurácias e número de regras no Experimento 4.

Discussões

- Neste experimento em particular, não foi possível obter resultados para muitas bases de dados. Para uma melhor identificação das bases de dados mais afetadas pelas variações em métodos de associação, elas foram separadas na Tabela 4.14 por uma linha divisória entre (parte superior) as binárias e as de múltiplas classes.
- Os métodos de associação Frequência e CD não geraram resposta para todas as bases de dados, sendo que a Frequência gerou um número menor de resultados e com acurácia inferior à proporcionada pelo CD. Este, por outro lado, não se mostrou eficaz para o caso de múltiplas classes. Uma razão possível é o compartilhamento de características por certas classes, o que, ao se fazer uso do CD, leva à consideração de apenas uma delas, fazendo com que algumas classes fiquem sem premissas. Este efeito é menos presente em bases binárias.

Como o objetivo é gerar sistemas interpretáveis, o problema foi então dividido em dois caminhos: uma configuração padrão para as bases de dados binárias e outra para as de múltiplas classes. Para as bases binárias, usa-se o Método de Associação CD (garante maior interpretabilidade); para as de múltiplas classes, o MQR.

4.1.5 Experimento 5 - Tipos de Funções de Pertinência e Filtro de Suporte

Avaliou-se, neste experimento, a influência dos tipos de Funções de Pertinência (até este ponto utilizou-se a Fortemente Particionada, também conhecida como Padrão) e do Filtro de Suporte (até agora foi usada a cardinalidade). As seguintes combinações foram avaliadas:

- Fortemente Particionada (Padrão) - Cardinalidade
- Tukey - Cardinalidade
- Fortemente Particionada (Padrão) - Frequência
- Tukey - Frequência.

A partir das configurações encontradas (para as bases binárias e de múltiplas classes) no Experimento 4, a Figura 4.6 apresenta as configurações avaliadas neste experimento.

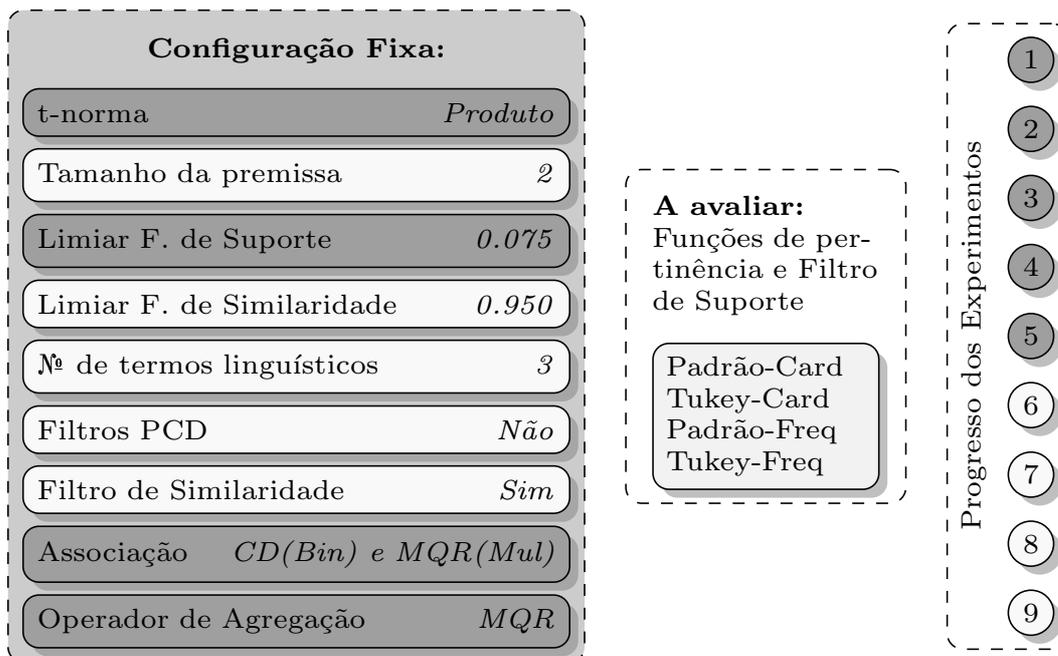


Figura 4.6: Experimento de funções de pertinência e tipo de suporte

4.1.5.1 Bases Binárias

As acurácias e o número de regras para as configurações do Experimento 5 são apresentados na Tabela 4.15. Os resultados do teste de Friedman e Holm são apresentadas na Tabela 4.16 e Tabela 4.17, respectivamente.

Base de dados	Acurácia				Número de regras				
	Cardinalidade		Frequência		Cardinalidade		Frequência		
	Padrão	Tukey	Padrão	Tukey	Padrão	Tukey	Padrão	Tukey	
1	appendicitis	87.91	87.91	86.09	84.18	4.70	4.30	4.30	3.70
2	australian	85.51	85.51	85.94	85.51	14.10	15.50	11.60	15.00
3	banana	58.75	59.75	58.75	59.55	3.00	4.00	3.00	3.40
4	bupa		64.28		64.05		7.00		5.30
5	crx	86.30	86.15	86.45	85.85	14.00	14.70	12.20	13.30
6	haberman				74.18				3.20
7	heart	83.33	83.33	83.70	82.96	13.90	14.70	12.50	12.80
8	hepatitis	83.42	86.23	87.24	82.47	9.10	11.90	9.90	9.80
9	magic	74.00	72.49	74.20	72.70	4.40	5.40	4.20	4.40
10	monk-2	96.81	96.81	96.81	96.81	7.40	7.40	7.40	7.40
11	phoneme	71.30	73.11	71.63	73.50	3.30	5.00	3.20	4.20
12	pima	75.67	73.58	74.62	73.58	6.00	6.90	5.00	6.50
13	saheart	72.52	72.94	72.75	69.45	7.20	10.10	6.60	8.60
14	titanic	77.60	77.60	77.60	77.60	10.60	10.60	10.60	10.60
15	wisconsin	94.16	94.76	93.14	95.04	11.30	9.00	9.30	9.40

Tabela 4.15: Acurácia e número de regras do Experimento 5 nas bases binárias.

i	Configuração	Posto
3	Tukey - Frequência	2.9231
2	Fort. Part. - Cardinalidade	2.5769
1	Tukey - Cardinalidade	2.3846
0	Fort. Part. - Frequência	2.1154

Teste	p-valor
Friedman	<0.4402
Iman e Davepor	<0.4543

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
Padrão - Cardinalidade	1.5951	0.1107	0.0167	Não
Tukey - Cardinalidade	0.9114	0.3621	0.0250	Não
Padrão - Frequência	0.5317	0.5949	0.0500	Não

Tabela 4.16: Resultados do teste de Friedman e Holm para acurácias no Experimento 5 nas bases binárias.

i	Configuração	Posto
3	Tukey - Cardinalidade	3.4231
2	Tukey - Frequência	2.500
1	Fort. Part. - Cardinalidade	2.4615
0	Fort. Part. - Frequência	1.6154

Teste	p-valor
Friedman	<0.0052
Iman e Davepor	<0.0023

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
Tukey - Cardinalidade	3.5699	0.0004	0.0167	Sim
Tukey - Frequência	1.7470	0.0806	0.0250	Não
Padrão - Cardinalidade	1.6710	0.0947	0.0500	Não

Tabela 4.17: Resultados do teste de Friedman para o número de regras do Experimento 5 nas bases binárias.

Discussões

- Quanto à acurácia, a configuração **Fortemente Particionada (Padrão) - Frequência** obteve o melhor ranking (menor posto), mas não se mostrou substancialmente superior às outras configurações (Tabela 4.17).
- Em termos do número de regras, a configuração **Fortemente Particionada (Padrão) - Frequência** também obteve o melhor ranking (menor posto), mas sem ser substancialmente superior às outras configurações (Tabela 4.17).
- A configuração **Tukey - Frequência** foi a escolhida, pois foi a única que gerou respostas para todas as bases de dados binárias e não se mostrou substancialmente inferior à configuração **Fortemente Particionada (Padrão) - Frequência**, seja quanto a acurácia ou ao número de regras.

4.1.5.2 Múltiplas classes

As acurácias e o número de regras obtidos das configurações do Experimento 5 são apresentados na Tabela 4.18. O resultado do teste de Friedman e Holm para as acurácias é apresentado na Tabela 4.19; para o número de regras, na Tabela 4.20.

Base de dados	Acurácia				Número de regras			
	Cardinalidade		Frequência		Cardinalidade		Frequência	
	Padrão	Tukey	Padrão	Tukey	Padrão	Tukey	Padrão	Tukey
1 balance	89.76	89.76	89.76	89.76	24.90	24.90	24.90	24.90
2 cleveland	58.96	58.00	57.92	57.95	50.30	51.20	52.60	52.70
3 contraceptive	52.55	55.54	52.89	55.81	29.60	31.70	29.70	30.90
4 ecoli	73.79	69.94	71.43	70.53	48.90	48.60	48.90	47.10
5 glass	60.50	56.88	61.86	54.94	24.40	29.80	24.80	28.50
6 hayes-roth	70.62	70.62	70.62	70.62	11.20	11.20	11.20	11.20
7 iris	95.33	94.00	94.67	90.00	6.00	5.00	6.30	6.10
8 newthyroid	86.58	89.83	85.17	90.76	8.60	7.40	8.60	7.00
9 page-blocks	89.80	93.31	91.43	93.81	15.70	23.50	16.60	18.30
10 tae	43.12	44.46	44.46	49.12	17.80	17.60	16.90	17.70
11 thyroid	92.31	93.39	11.04	93.57	10.00	14.40	10.00	10.10
12 vehicle	56.27	56.75	57.56	57.58	18.00	20.20	18.50	18.90
13 vowel	45.15	45.35	46.87	48.48	67.10	69.90	64.50	65.20
14 wine	92.65	92.09	95.46	91.54	10.60	9.70	10.50	9.50
15 winequality-red	57.29	57.35	56.66	57.04	28.10	41.80	33.70	46.90
16 yeast	50.40	52.63	54.18	53.17	31.60	46.20	40.50	51.60

Tabela 4.18: Acurácia e número de regras do Experimento 5 nas múltiplas classes.

i	Configuração	Posto
3	Padrão - Cardinalidade	2.7500
2	Tukey - Cardinalidade	2.5312
1	Padrão - Frequência	2.5312
0	Tukey - Frequência	2.1875

Teste	p-valor
Friedman	<0.6694
Iman e Davepor	<0.6823

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
Padrão - Cardinalidade	1.3224	0.2178	0.0167	Não
Tukey - Cardinalidade	0.7531	0.4514	0.0250	Não
Padrão - Frequência	0.7531	0.4514	0.0500	Não

Tabela 4.19: Resultados do teste de Friedman e Holm para acurácias do Experimento 5 nas bases de múltiplas classes.

i	Configuração	Posto
3	Tukey - Cardinalidade	2.8750
2	Tukey - Frequência	2.6875
1	Padrão - Frequência	2.3438
0	Padrão - Cardinalidade	2.0938

Teste	p-valor
Friedman	<0.3199
Iman e Davepor	<0.3272

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
Tukey - Cardinalidade	1.3948	0.1631	0.0167	Não
Tukey - Frequência	0.8635	0.3879	0.0250	Não
Padrão - Frequência	0.3985	0.6902	0.0500	Não

Tabela 4.20: Resultados do teste de Friedman para o número de regras do Experimento 5 nas bases de múltiplas classes.

Discussões

- Na acurácia, a configuração **Tukey - Frequência** obteve o melhor ranking (menor posto) mas sem vantagem significativa com respeito às outras configurações (ver Tabela 4.19).
- No número de regras, a configuração **Fortemente Particionada (Padrão) - Cardinalidade** obteve o melhor ranking (menor posto) mas sem vantagem significativa com respeito às outras configurações (ver Tabela 4.20).
- Foi escolhida a configuração **Tukey - Frequência** considerando o ranking das acurácias, dado que pelo número de regras foi indistinto.

4.1.6 Experimento 6 - Número de termos linguísticos

Avaliou-se a influência do número de termos linguísticos (usado até agora: 3), onde três valores são avaliados: 3, 5 e 7. A Figura 4.7 apresenta as configurações avaliadas neste experimento.

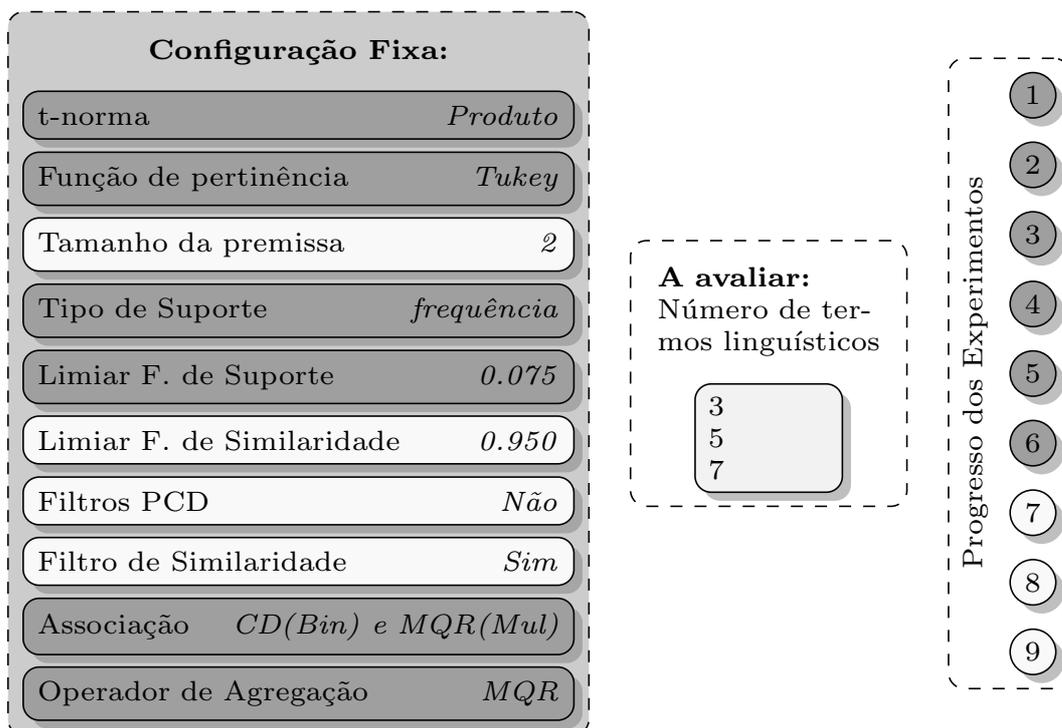


Figura 4.7: Experimento de número de termos linguísticos.

4.1.6.1 Bases Binárias

As acurácias e o número de regras obtidos nas configurações do Experimento 6 são apresentados na Tabela 4.21. Os resultados para o teste de Friedman e Holm são apresentados na Tabela 4.22 e na Tabela 4.23, respectivamente.

	Base de dados	Acurácia			Número de regras		
		P3	P5	P7	P3	P5	P7
1	appendicitis	84.18	77.45	82.27	3.70	6.00	7.40
2	australian	85.51	85.36	86.23	15.00	12.50	14.80
3	banana	59.55	70.26	65.85	3.40	6.00	5.20
4	bupa	64.05	59.43	59.62	5.30	10.30	11.00
5	crx	85.85	85.71	86.33	13.30	11.60	13.00
6	haberman	74.18	72.83	70.82	3.20	7.80	10.40
7	heart	82.96	81.85	81.48	12.80	15.20	14.50
8	hepatitis	82.47	88.33	88.91	9.80	11.00	11.20
9	magic	72.70	71.26	73.53	4.40	8.30	10.40
10	monk-2	96.81	96.81	96.59	7.40	7.40	7.40
11	phoneme	73.50	77.65	75.96	4.20	8.70	10.20
12	pima	73.58	75.17	73.20	6.50	8.50	9.00
13	saheart	69.45	68.43	66.00	8.60	11.20	10.00
14	titanic	77.60	77.60	77.60	10.60	10.60	10.60
15	wisconsin	95.04	94.44	95.78	9.40	10.50	11.30

Tabela 4.21: Acurácias e número de regras do Experimento 6 nas bases binárias.

i	Configuração	Posto
2	P5	2.9231
1	P7	2.3846
0	P3	2.1154

Teste	p-valor
Friedman	<0.6592
Iman e Davepor	<0.6741

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
P5	0.9129	0.3613	0.0250	Não
P7	0.4564	0.6481	0.0500	Não

Tabela 4.22: Resultados do teste de Friedman e Holm para acurácias nas bases binárias do Experimento 6.

i	Configuração	Posto
3	P7	2.5333
1	P5	2.0667
0	P3	1.400

Teste	p-valor
Friedman	<0.0077
Iman e Davepor	<0.0041

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
P7	3.1038	0.0019	0.0250	Sim
P5	1.8226	0.0679	0.0500	Não

Tabela 4.23: Resultados do teste de Friedman para o número de regras do Experimento 6 nas bases binárias.

Discussões

- No número de regras, a configuração P3 obteve o melhor ranking (menor posto), com vantagem significativa sobre P7 mas não sobre P5.
- Na acurácia, a configuração P3 obteve o melhor ranking (menor posto) mas sem vantagem significativa sobre as outras configurações.
- Escolheu-se a configuração P3 por obter o melhor ranking nos testes de Friedman e Holm para a acurácia e para o número de regras.

4.1.6.2 Múltiplas classes

Os resultados obtidos para as bases de múltiplas classes são apresentados na Tabela 4.24. As Tabelas 4.25 e 4.26 apresentam os resultados do Teste de Friedman e Holm .

	Base de dados	Acurácia			Número de regras		
		P3	P5	P7	P3	P5	P7
1	balance	89.76	69.43	69.43	24.90	35.90	51.90
2	cleveland	57.95	59.64	61.31	52.70	63.20	67.10
3	contraceptive	55.81	50.51	48.20	30.90	35.40	43.00
4	ecoli	70.53	73.85	73.23	47.10	76.00	100.50
5	glass	54.94	63.08	61.29	28.50	41.90	50.70
6	hayes-roth	70.62	70.62	70.62	11.20	11.20	11.20
7	iris	90.00	95.33	97.33	6.10	7.60	6.20
8	newthyroid	90.76	91.62	80.95	7.00	10.30	13.10
9	page-blocks	93.81	93.18	83.26	18.30	27.50	36.90
10	tae	49.12	45.79	51.08	17.70	20.10	20.70
11	thyroid	93.57	92.58	92.76	10.10	18.00	26.50
12	vehicle	57.58	61.59	55.43	18.90	28.40	34.70
13	vowel	48.48	62.63	62.73	65.20	94.00	109.10
14	wine	91.54	94.35	90.95	9.50	14.40	10.40
15	winequality-red	57.04	57.66	56.35	46.90	89.80	136.80
16	yeast	53.17	52.90	53.44	51.60	108.20	160.40

Tabela 4.24: Acurácias e número de regras do Experimento 6 nas bases de múltiplas classes.

i	Configuração	Posto
2	P7	2.0938
1	P3	2.0625
0	P5	1.8438

Teste	p-valor
Friedman	<0.7431
Iman e Davepor	<0.7551

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
P7	0.7071	0.4795	0.0250	Não
P3	0.6187	0.5361	0.0500	Não

Tabela 4.25: Resultados do teste de Friedman e Holm para acurácias nas bases de múltiplas classes do Experimento 6.

i	Configuração	Posto
2	P7	2.8125
1	P5	2.1250
0	P3	1.0625

Teste	p-valor
Friedman	<0.0001
Iman e Davenport	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
P7	4.9497	0.0001	0.0250	Sim
P5	3.0052	0.0027	0.0500	Sim

Tabela 4.26: Resultados do teste de Friedman para o número de regras nas bases de múltiplas classes do Experimento 6.

Discussões

- No que concerne ao número de regras, a configuração P3 obteve o melhor ranking (menor posto), com vantagem significativa sobre as outras configurações.
- Em termos de acurácia, a configuração P5 obteve o melhor ranking (menor posto), mas sem vantagem significativa sobre P3 nem P7.
- Escolheu-se a configuração P3 por gerar uma base de regras mais compacta.

4.1.7 Experimento 7 - Seleção de Filtros

O experimento 7 destinou-se a investigar a influência dos filtros PCD e Similaridade, onde quatro combinações foram consideradas:

- CF1: Filtro de Similaridade
- CF2: PCD e Filtro de Similaridade
- CF3: Nenhum destes filtros
- CF4: PCD

A Figura 4.8 apresenta as configurações investigadas.

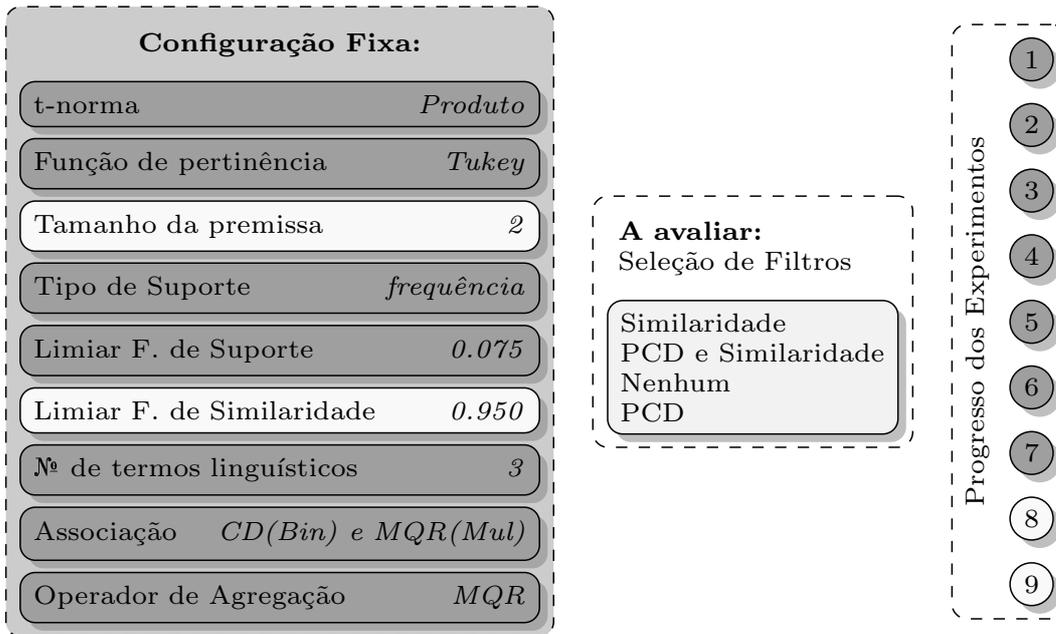


Figura 4.8: Experimento com uso dos Filtros PCD e Similaridade.

4.1.7.1 Bases Binárias

São apresentadas na Tabela 4.27 as acurácias e o número de regras para as configurações do Experimento 7. Os resultados do teste de Friedman e Holm são apresentados na Tabela 4.28 e na Tabela 4.29.

	Base de dados	Acurácia				Número de regras			
		CF1	CF2	CF3	CF4	CF1	CF2	CF3	CF4
1	appendicitis	84.18	86.09	85.18	85.18	3.70	3.50	3.50	3.50
2	australian	85.51	86.09	85.94	85.94	15.00	15.60	16.90	16.70
3	banana	59.55	59.55	59.55	59.55	3.40	3.40	3.40	3.40
4	bupa	64.05	64.42	64.42	64.14	5.30	5.70	5.70	5.60
5	crx	85.85	85.70	85.70	85.24	13.30	14.40	18.10	18.00
6	haberman	74.18	74.18	74.18	74.18	3.20	3.20	3.20	3.20
7	heart	82.96	80.37	83.70	83.70	12.80	13.70	14.50	14.40
8	hepatitis	82.47	83.42	84.32	84.32	9.80	10.30	10.50	10.50
9	magic	72.70	73.05	72.84	72.78	4.40	4.00	4.20	4.30
10	monk-2	96.81	96.81	96.59	96.59	7.40	7.40	7.40	7.40
11	phoneme	73.50	73.50	73.50	73.50	4.20	4.20	4.20	4.20
12	pima	73.58	73.71	72.80	73.19	6.50	6.00	6.50	6.10
13	saheart	69.45	70.76	70.76	70.55	8.60	8.30	8.30	8.20
14	titanic	77.60	77.60	77.60	77.60	10.60	10.60	7.80	7.80
15	wisconsin	95.04	95.04	95.49	95.34	9.40	9.20	8.80	9.10

Tabela 4.27: Acurácia e número de regras para o Experimento 7 nas bases binárias.

i	Configuração	Posto
3	CF1	3.0000
2	CF4	2.6333
1	CF3	2.3000
0	CF2	2.0667

Teste	p-valor
Friedman	<0.2159
Iman e Davepor	<0.2182

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
CF1	1.9799	0.0477	0.0167	Não
CF4	1.2021	0.2293	0.0250	Não
CF3	0.4950	0.6206	0.0500	Não

Tabela 4.28: Resultados do teste de Friedman e Holm para acurácia nas bases binárias do Experimento 7.

i	Configuração	Posto
3	CF3	2.7667
2	CF1	2.5333
1	CF4	2.4000
0	CF2	2.3000

Teste	p-valor
Friedman	<0.7771
Iman e Davepor	<0.7888

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
CF3	0.9899	0.3222	0.0167	Não
CF1	0.4950	0.6206	0.0250	Não
CF4	0.2121	0.8320	0.0500	Não

Tabela 4.29: Resultados do teste de Friedman para o número de regras do Experimento 7 nas bases binárias.

Discussões

Nos testes de Friedman e Holm para acurácia e número de regras, nenhuma configuração mostrou-se claramente superior às outras. Por ter apresentado a melhor qualificação, escolheu-se a configuração CF2: Filtro PCD e Similaridade.

4.1.7.2 Múltiplas classes

As acurácias e número de regras para configurações do Experimento 7 são apresentadas na Tabela 4.30. As comparações de acurácia e de número de regras pelo Teste de Friedman e Holm são apresentadas na Tabela 4.31 e na Tabela 4.32, respectivamente.

	Base de dados	Acurácia				Número de regras			
		CF1	CF2	CF3	CF4	CF1	CF2	CF3	CF4
1	balance	89.76	89.44	89.76	89.44	24.90	17.80	24.90	17.80
2	cleveland	57.95	57.21	58.62	57.21	52.70	32.20	49.60	32.30
3	contraceptive	55.81	45.55	56.15	45.55	30.90	17.10	30.70	17.10
4	ecoli	70.53	60.12	70.24	60.12	47.10	38.90	52.30	38.90
5	glass	54.94	50.58	57.33	50.58	28.50	13.80	28.60	13.80
6	hayes-roth	70.62	71.25	70.62	71.25	11.20	9.50	11.20	9.50
7	iris	90.00	94.00	94.00	94.00	6.10	4.00	4.90	4.00
8	newthyroid	90.76	90.71	91.19	90.71	7.00	7.30	7.30	7.30
9	page-blocks	93.81	93.64	93.82	93.77	18.30	18.00	18.70	18.60
10	tae	49.12	39.04	48.46	39.04	17.70	14.80	17.80	14.80
11	thyroid	93.57	93.62	93.56	93.49	10.10	13.20	12.90	16.10
12	vehicle	57.58	41.86	56.63	41.15	18.90	10.00	20.60	9.80
13	vowel	48.48		47.98		65.20		63.60	
14	wine	91.54	94.35	91.54	94.35	9.50	8.50	9.80	8.50
15	winequality-red	57.04	45.71	56.78	45.71	46.90	24.70	47.00	24.70
16	yeast	53.17	30.72	52.43	34.02	51.60	10.00	50.00	10.00

Tabela 4.30: Acurácia e número de regras do Experimento 7 nas bases de múltiplas classes.

Discussões

- Relativamente à acurácia, a configuração com melhor ranking foi a que não usou os filtros do PCD e de Similaridade, embora não tenha se mostrado nitidamente superior àquela que faz uso da Similaridade apenas (veja Tabela 4.31). Destaca-se que a configuração com apenas Similaridade obteve a melhor acurácia para um maior número de bases de dados (8).
- Para número de regras, as configurações com Filtro PCD geraram poucas regras mas nem sempre produziram resultados para todas as bases de dados – como ocorre com a base de dados **vowel**.
- Baseados nas duas métricas, a configuração escolhida foi aquela que utilizou somente o Filtro de Similaridade.

i	Configuração	Posto
3	CF4	3.1333
2	CF2	3.0000
1	CF1	1.9667
0	CF3	1.9

Teste	p-valor
Friedman	<0.0086
Iman e Davepor	<0.0052

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
CF4	2.6163	0.0089	0.0250	Sim
CF2	2.3335	0.0196	0.0250	Sim
CF1	0.1414	0.8875	0.0500	Não

Tabela 4.31: Resultados do teste de Friedman e Holm para acurácias nas bases de múltiplas classes do Experimento 7.

i	Configuração	Posto
3	CF3	3.4667
2	CF1	3.0000
1	CF4	1.8667
0	CF2	1.6667

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
CF3	3.8184	0.0001	0.0167	Sim
CF1	2.8284	0.0046	0.0250	Sim
CF4	0.4243	0.6714	0.0500	Não

Tabela 4.32: Resultados do teste de Friedman para o número de regras nas bases de múltiplas classes do Experimento 7.

4.1.8 Experimento 8 - Tamanho da premissa

Investigou-se agora a influência do tamanho das premissas, uma vez que, até agora, o tamanho usado foi sempre 2. Tamanhos testados:

- Tamanho 2
- Tamanho 3
- Tamanho 2 com uso do Operador de Negação

A partir das configurações encontradas no Experimento 7, a Figura 4.9 apresenta as configurações do presente experimento.

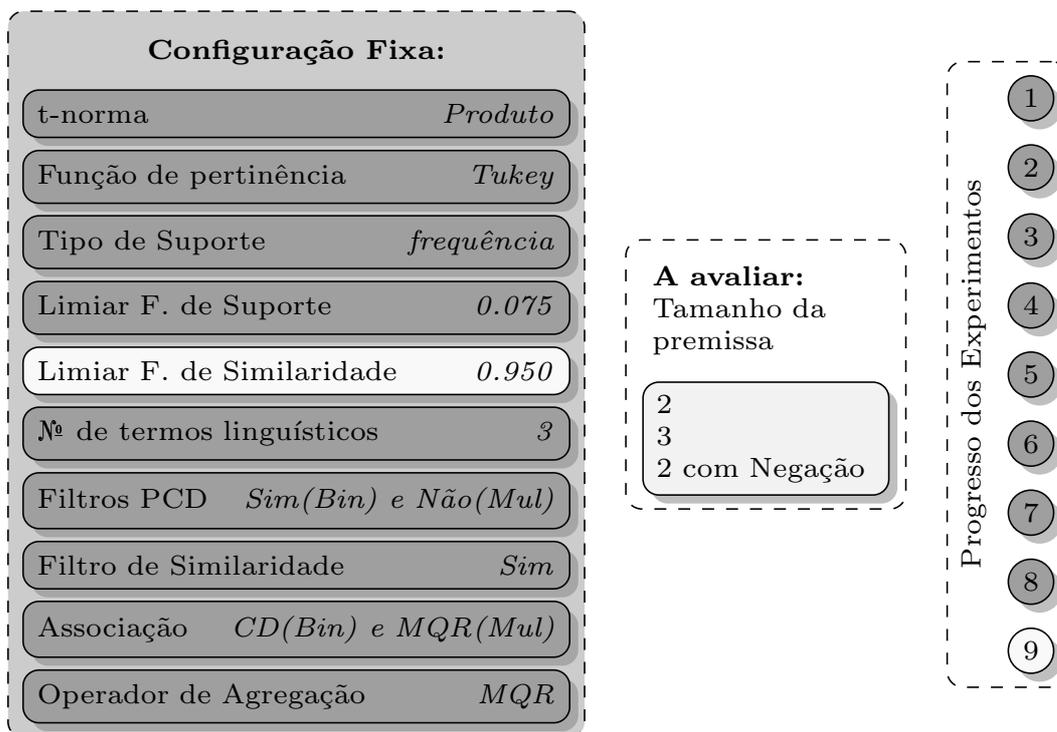


Figura 4.9: Experimento com o tamanho da premissa.

4.1.8.1 Bases Binárias

Os resultados das configurações do Experimento 8 são apresentados na Tabela 4.33. Já os resultados do Teste de Friedman e Holm, para acurácia e número de regras, são apresentados na Tabela 4.34 e na Tabela 4.35, respectivamente.

	Base de dados	Acurácia			Número de regras		
		2	3	2Neg	2	3	2Neg
1	appendicitis	86.09	83.18	85.18	3.50	3.90	5.40
2	australian	86.09	83.91	85.65	15.60	10.00	17.50
3	banana	59.55	59.55	60.08	3.40	3.40	4.30
4	bupa	64.42	63.02	65.84	5.70	4.40	6.60
5	crx	85.70	82.72	85.86	14.40	11.20	15.60
6	haberman	74.18	71.89	73.53	3.20	3.10	5.20
7	heart	80.37	79.26	82.22	13.70	10.70	17.10
8	hepatitis	83.42	85.32	82.66	10.30	9.60	11.10
9	magic	73.05	71.62	79.59	4.00	3.60	7.90
10	monk-2	96.81	95.90	97.27	7.40	6.30	9.90
11	phoneme	73.50	73.37	72.13	4.20	2.80	5.10
12	pima	73.71	73.44	73.84	6.00	5.00	7.30
13	saheart	70.76	66.20	72.08	8.30	6.30	11.20
14	titanic	77.60	77.65	77.60	10.60	8.60	9.20
15	wisconsin	95.04	94.92	96.79	9.20	6.70	13.80

Tabela 4.33: Acurácias das bases de dados no Experimento 8 nas bases binárias.

i	Configuração	Posto
2	T3	2.6333
1	T2	1.8
0	T2Neg	1.5667

Teste	p-valor
Friedman	<0.0089
Iman e Davepor	<0.0051

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
T3	2.9212	0.0035	0.0250	Sim
T2	0.6390	0.5228	0.0500	Não

Tabela 4.34: Resultados do teste de Friedman e Holm para acurácias Experimento 8 nas bases binárias.

i	Configuração	Posto
2	T2Neg	2.9333
1	T2	1.9667
0	T3	1.1000

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
T2Neg	5.0208	0.0001	0.0250	Sim
T2	2.3735	0.0176	0.0500	Sim

Tabela 4.35: Resultados do teste de Friedman para o número de regras do Experimento 8 nas bases binárias.

Discussões

- Para acurácia, o tamanho 2 com uso do operador negação foi a melhor configuração, manifestando-se a grande influência do operador de negação na acurácia do modelo em comparação a se aumentar o tamanho da premissa.
- De acordo ao teste de Friedman e Holm para o número de regras, premissas de tamanho 3 produziram resultados substancialmente superiores aos das outras configurações.
- Não foi possível decidir entre uma configuração e outra. Levando em consideração o custo computacional (Tabela 4.36), foi escolhida a configuração de tamanho 2 com negação que, adicionalmente, forneceu a melhor acurácia no ranking de Friedman.

	Base de dados	Tamanho da premissa		
		2	3	2Neg
1	appendicitis	1.24	8.10	3.60
2	australian	20.14	1222.38	262.21
3	banana	1.53	1.53	1.94
4	bupa	1.45	9.23	5.06
5	crx	22.19	1976.38	166.66
6	haberman	0.34	0.53	0.79
7	heart	7.99	591.46	88.12
8	hepatitis	6.20	640.85	16.10
9	magic	474.76	47324.54	2501.16
10	monk-2	1.00	2.69	4.80
11	phoneme	9.53	64.52	55.16
12	pima	6.20	233.58	75.70
13	saheart	4.90	265.93	44.36
14	titanic	0.41	0.47	0.57
15	wisconsin	5.94	191.02	89.30

Tabela 4.36: Tempos empregados (segundos) pelas configurações no Experimento 8 nas bases binárias.

4.1.8.2 Múltiplas classes

As acurácias e o número de regras das configurações do Experimento 8 são apresentadas na Tabela 4.37. As comparações da acurácia e do número de regras são analisadas nas Tabelas 4.38 e 4.39, respectivamente.

	Base de dados	Acurácia			Número de regras		
		2	3	2Neg	2	3	2Neg
1	balance	89.76	82.90	90.08	24.90	23.80	25.00
2	cleveland	57.95	55.90	59.62	52.70	57.70	51.30
3	contraceptive	55.81	55.27	56.49	30.90	33.70	42.40
4	ecoli	70.53	72.63	75.60	47.10	46.80	38.80
5	glass	54.94	62.84	55.49	28.50	28.80	28.30
6	hayes-roth	70.62	70.62	74.38	11.20	11.20	20.60
7	iris	90.00	94.00	96.67	6.10	4.60	6.90
8	newthyroid	90.76	91.65	90.30	7.00	7.20	6.60
9	page-blocks	93.81	94.12	93.53	18.30	20.90	17.50
10	tae	49.12	48.46	46.54	17.70	18.80	23.00
11	thyroid	93.57	93.54	93.31	10.10	11.30	13.80
12	vehicle	57.58	55.44	59.46	18.90	20.60	21.60
13	vowel	48.48	49.70	49.09	65.20	68.00	59.40
14	wine	91.54	92.65	94.90	9.50	9.00	9.70
15	winequality-red	57.04	57.35	58.79	46.90	49.10	38.60
16	yeast	53.17	53.51	51.95	51.60	51.50	47.70

Tabela 4.37: Acurácias e número de regras das bases de múltiplas classes no Experimento 8.

i	Configuração	Posto
2	T2	2.2812
1	T3	1.9688
0	T2Neg	1.7500

Teste	p-valor
Friedman	<0.3196
Iman e Davapor	<0.3298

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
T2	1.5026	0.1329	0.0250	Não
T3	0.6187	0.5361	0.0500	Não

Tabela 4.38: Resultados do teste de Friedman e Holm para acurácias nas bases de múltiplas classes do Experimento 8.

i	Configuração	Posto
2	T3	2.1562
1	T2Neg	2.0000
0	T2	1.8438

Teste	p-valor
Friedman	<0.6766
Iman e Davepor	<0.6902

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
T3	0.8839	0.3768	0.0250	Não
T2Neg	0.4419	0.6585	0.0500	Não

Tabela 4.39: Resultados do teste de Friedman para o número de regras nas bases de múltiplas classes do Experimento 8.

Discussões

Não foi encontrada uma configuração substancialmente superior, seja em termos de número de regras ou de acurácia, nos testes de Friedman e Holm. Com base nos rankings das duas métricas, foi escolhida a configuração de tamanho 2 com negação: menor posto em acurácia e segundo menor posto no número de regras. Além disso, na Tabela 4.40 pode-se observar que o custo computacional para a configuração de tamanho 2 é menor do que para o tamanho 3.

Base de dados	Tamanho da premissa		
	2	3	2Neg
1 balance	0.97	1.75	3.80
2 cleveland	15.08	769.40	124.08
3 contraceptive	8.45	74.89	120.77
4 ecoli	4.69	14.10	6.69
5 glass	8.88	186.37	29.93
6 hayes-roth	0.61	0.73	2.53
7 iris	1.03	1.83	1.60
8 newthyroid	1.80	5.15	3.33
9 page-blocks	124.61	11829.29	883.10
10 tae	0.73	0.83	3.79
11 thyroid	82.50	4198.63	311.88
12 vehicle	208.79	57326.09	2136.79
13 vowel	56.80	3675.81	1045.40
14 wine	16.22	1256.08	110.32
15 winequality-red	62.47	3980.35	668.24
16 yeast	10.51	108.41	40.92

Tabela 4.40: Tempos empregados (segundos) pelas configurações no Experimento 8 nas bases de múltiplas classes.

4.1.9 Experimento 9 - Limiar do Filtro de Similaridade

Finalmente, o Experimento 9 buscou definir um valor de limiar para o Filtro de Similaridade (até agora: 0.95). Os limiares avaliados neste experimento foram 0.925, 0.950 e 0.975.

A Figura 4.10 apresenta as configurações deste experimento.

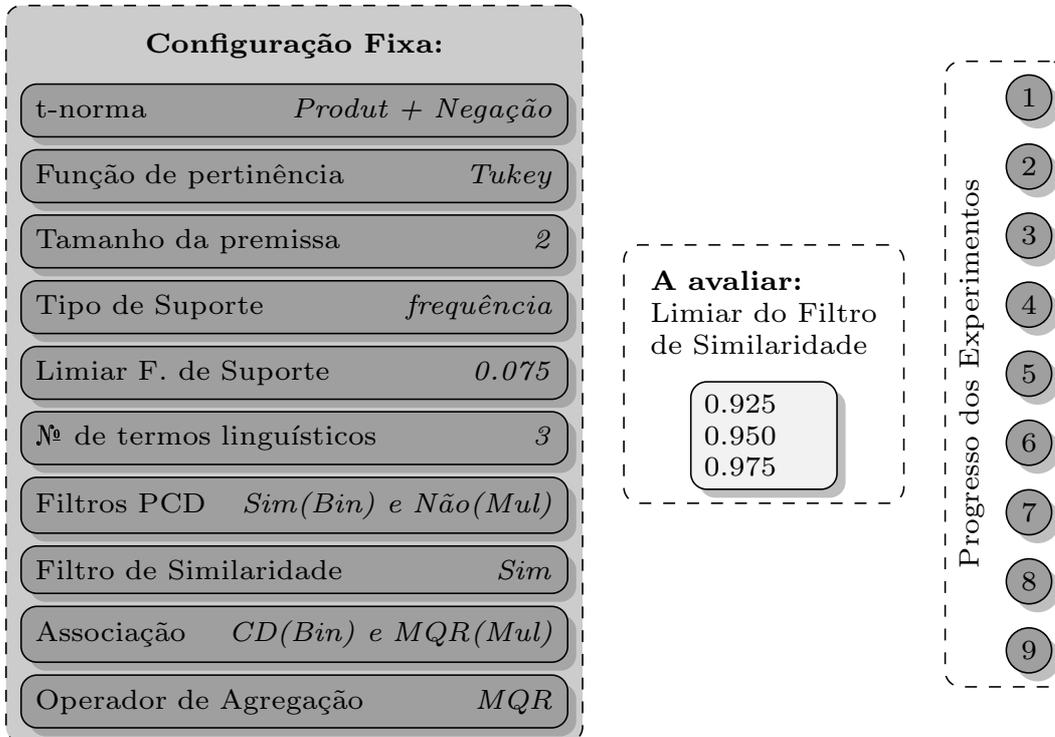


Figura 4.10: Experimento do Limiar do Filtro de Similaridade.

4.1.9.1 Bases Binárias

As acurácias e número de regras para as configurações do Experimento 9 são apresentadas na Tabela 4.41. As análises mediante o Teste de Friedman e Holm são apresentadas Tabela 4.42 e na Tabela 4.43.

	Base de dados	Acurácia			Número de regras		
		0.925	0.950	0.975	0.925	0.950	0.975
1	appendicitis	88.00	85.18	86.09	5.30	5.40	5.40
2	australian	85.65	85.65	85.36	18.30	17.50	18.20
3	banana	60.08	60.08	60.08	4.30	4.30	4.30
4	bupa	64.96	65.84	65.82	6.70	6.60	7.40
5	crx	86.17	85.86	86.91	14.60	15.60	14.80
6	haberman	73.53	73.53	73.53	5.20	5.20	5.20
7	heart	83.33	82.22	82.96	17.00	17.10	16.90
8	hepatitis	80.32	82.66	82.51	10.40	11.10	10.30
9	magic	77.34	79.59	80.02	7.10	7.90	8.30
10	monk-2	97.27	97.27	97.27	9.90	9.90	9.90
11	phoneme	71.91	72.13	72.30	5.70	5.10	5.80
12	pima	73.32	73.84	74.23	7.30	7.30	7.80
13	saheart	71.22	72.08	72.52	11.70	11.20	11.80
14	titanic	77.60	77.60	77.60	9.20	9.20	9.40
15	wisconsin	96.65	96.79	96.06	13.10	13.80	12.70

Tabela 4.41: Acurácias e número de regras para as bases binárias no Experimento 9.

i	Configuração	Posto
2	0.925	2.2333
1	0.950	1.9667
0	0.975	1.8000

Teste	p-valor
Friedman	<0.4884
Iman e Davepor	<0.5039

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
0.925	1.186732	0.2353	0.0250	Não
0.950	0.456435	0.6481	0.5000	Não

Tabela 4.42: Resultados do teste de Friedman e Holm para acurácias do Experimento 9 nas bases binárias.

i	Configuração	Posto
2	0.975	2.2333
1	0.950	1.9667
0	0.925	1.8000

Teste	p-valor
Friedman	<0.4884
Iman e Davepor	<0.5039

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
0.975	1.1867	0.2353	0.0250	Não
0.950	0.4564	0.6481	0.0500	Não

Tabela 4.43: Resultados do teste de Friedman para o número de regras do Experimento 9 nas bases binárias.

Discussões

- Para acurácia, o limiar 0.975 foi a configuração de melhor ranking (menor posto), mas sem mostrar superioridade nítida em relação às outras configurações.
- No número de regras, o limiar 0.925 foi a configuração de melhor ranking (menor posto), mas também sem uma clara superioridade.
- Escolheu-se limiar 0.950 por ser este um valor intermediário para as duas métricas usadas.

4.1.9.2 Múltiplas classes

Os resultados do Experimento 9 são apresentados na Tabela 4.44. Os resultados do teste de Friedman e Holm são apresentados nas Tabelas 4.38 e 4.39.

Base de dados	Acurácia			Número de regras		
	0.925	0.950	0.975	0.925	0.950	0.975
1 balance	90.08	90.08	90.08	25.00	25.00	25.00
2 cleveland	58.59	59.62	58.95	54.90	51.30	54.60
3 contraceptive	56.29	56.49	55.74	40.60	42.40	40.60
4 ecoli	75.60	75.60	74.99	38.00	38.80	38.50
5 glass	54.78	55.49	56.14	30.50	28.30	29.30
6 hayes-roth	74.38	74.38	73.12	20.90	20.60	20.70
7 iris	96.00	96.67	96.00	6.50	6.90	6.70
8 newthyroid	89.83	90.30	91.67	6.70	6.60	7.20
9 page-blocks	93.13	93.53	93.46	18.50	17.50	18.90
10 tae	49.75	46.54	49.12	21.90	23.00	22.50
11 thyroid	93.40	93.31	92.94	11.10	13.80	12.80
12 vehicle	60.54	59.46	61.37	20.30	21.60	20.90
13 vowel	49.09	49.09	49.39	60.90	59.40	60.80
14 wine	96.60	94.90	97.19	9.90	9.70	9.60
15 winequality-red	58.04	58.79	58.79	41.80	38.60	39.50
16 yeast	52.29	51.95	53.98	49.00	47.70	48.10

Tabela 4.44: Acurácias e número de regras das bases de múltiplas classes no Experimento 9.

i	Configuração	Posto
2	0.925	2.1875
1	0.950	1.9688
0	0.975	1.8438

Teste	p-valor
Friedman	<0.6161
Iman e Davepor	<0.6306

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
0.925	0.9723	0.3309	0.0250	Não
0.950	0.3536	0.7237	0.0500	Não

Tabela 4.45: Resultados do teste de Friedman e Holm para acurácias Experimento 9 nas bases de múltiplas classes.

i	Configuração	Posto
2	0.925	2.0938
1	0.975	2.0312
0	0.950	1.8750

Teste	p-valor
Friedman	<0.8162
Iman e Davepor	<0.8256

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
0.925	0.6187	0.5361	0.0250	Não
0.975	0.4419	0.6585	0.0500	Não

Tabela 4.46: Resultados do teste de Friedman regras para o número de regras do Experimento 9 nas bases de múltiplas classes.

Discussões

- Em termos de acurácia, o limiar 0.975 apresentou o menor posto, mas sem uma superioridade nítida.
- Para o número de regras, o limiar 0.950 foi o melhor rankeado (menor posto), mas também sem ser substancialmente superior.
- Escolheu-se o limiar 0.950: segundo rankeado na acurácia e primeiro no número de regras.

4.1.10 Resumo - Parâmetros das configuração obtidas

Foram apresentadas as principais análises empíricas do modelo AutoFIS-Class. Os experimentos possibilitaram avaliar as influências dos diferentes parâmetros do modelo proposto. O uso da negação clássica possibilitou o aprimoramento dos resultados em termos de acurácia e redução da complexidade da base de regras fuzzy. O formato triangular Tukey possibilitou a obtenção de resultados para muitas bases de dados. A utilidade do filtro PCD foi comprovado nas bases binárias, gerando bases de regras de maior compacidade sem comprometer significativamente a acurácia.

Nos método de Associação, o MQR proporcionou boa acurácia para múltiplas classes. Em contrapartida, a interpretabilidade não é garantida e é observado um incremento do número de regras. Nas bases binárias, o CD não gerou o efeito negativo observado em algumas bases de múltiplas classes.

Na Agregação, o MQR possibilitou ganhos em acurácia e interpretabilidade, apesar do aumento do tempo computacional envolvido. Embora se tenha verificado que o IntMQR possibilitou ganhos expressivos de acurácia em várias bases de dados, o número de regras gerado cresceu substancialmente. Por isso a opção pelo MQR.

As configurações finais para as bases de dados binárias e de múltiplas classe são expostas na Tabela 4.47.

	Bases Binárias	Bases Múltiplas
Formato da função de pertinência	tukey	tukey
No funções de pertinência	3	3
t-norma	Prod	Prod
Negação	Sim	Sim
Tamanho da premissa	2	2
Tipo de Suporte	frequência	frequência
Limiar de Suporte (ε_{sup})	0.075	0.075
Limiar de Similaridade (ε_{sim})	0.950	0.950
Filtro PCD	Sim	Não
Filtro de Similaridade	Sim	Sim
Associação	CD	MQR
Agregação	MQR	MQR

Tabela 4.47: Configurações empíricas do modelo AutoFIS-Class.

4.2 Extensão do modelo AutoFIS-Class para bases de dados de dimensão maior em atributos

Conforme indicado na Seção 4.1, a análise empírica foi realizada com um conjunto de bases de dados menor, devido ao alto custo computacional associado ao número de premissas geradas na etapa da Formulação do modelo AutoFIS-Class com bases com uma grande número de atributos (>25). Assim, de forma a possibilitar o uso do modelo em bases maiores, apresenta-se uma proposta de extensão do AutoFIS-Class. Levando em consideração que o modelo é determinístico, aplica-se um tipo de *bagging* [47], criando-se vários classificadores menores com subconjuntos da base de dados original – conjunto menor de atributos selecionados aleatoriamente. Cada um dos classificadores usará a mesma configuração definida na Seção anterior. Finalmente, as premissas obtidas de cada um dos classificadores menores são usadas para agregar o modelo AutoFIS-Class final.

4.2.1 Bases Binárias

A Tabela 4.49 apresenta o comportamento da extensão do AutoFIS-Class para bases binárias com muitos atributos (Tabela 4.1). Foram avaliadas quatro configurações, com números distintos de classificadores (25, 50, 75, 100). Nota-se que, em alguns casos, o desvio padrão é relativamente alto, podendo sugerir a necessidade de um maior número de classificadores ou que a configuração utilizada não é a mais adequada. A taxa de sucesso dos classificadores menores para gerar premissas em todas as classes é mostrado na Tabela 4.48. Em função desses resultados optou-se pela configuração com 100 classificadores menores.

Dataset	Número de classificadores			
	25	50	75	100
german	17.9	35.0	53.8	78.5
ionosphere	25.0	50.0	75.0	100.0
ring	25.0	50.0	75.0	100.0
sonar	25.0	50.0	75.0	100.0
spambase	23.9	46.0	71.0	94.7
spectfheart	11.3	22.4	31.7	46.5
twonorm	25.0	50.0	75.0	100.0
wdbc	25.0	50.0	75.0	100.0

Tabela 4.48: Taxa de sucesso dos classificadores menores nas bases binárias.

		Número de classificadores			
		Dataset	25	50	75
Ac	german	72.50	72.10	72.50	73.30
	ionosphere	82.93	84.93	83.79	84.36
	ring	50.51	51.30	50.50	56.12
	sonar	74.95	74.93	76.40	79.21
	spambase	88.42	89.25	89.73	89.80
	spectfheart	79.42	79.42	79.77	79.81
	twonorm	94.53	94.41	94.57	94.53
	wdbc	92.96	92.43	92.61	94.37
Std _{Ac}	german	2.80	1.64	1.86	1.85
	ionosphere	5.75	6.01	5.61	5.01
	ring	0.05	1.64	1.03	7.90
	sonar	8.97	9.85	10.00	13.16
	spambase	1.86	1.26	1.05	0.95
	spectfheart	1.66	2.87	2.47	3.20
	twonorm	1.28	1.07	1.16	1.73
	wdbc	3.44	3.45	4.25	2.93
Nr	german	27.0	30.3	34.6	38.3
	ionosphere	13.6	14.2	14.2	17.2
	ring	22.3	26.7	26.8	30.4
	sonar	15.7	15.6	15.2	16.8
	spambase	20.8	21.2	24.3	22.8
	spectfheart	9.2	10.5	10.7	12.1
	twonorm	29.2	30.1	32.0	31.9
	wdbc	6.8	7.3	8.0	8.8

Tabela 4.49: Extensão do modelo AutoFIS em bases binárias.

4.2.2 Múltiplas classes

Da mesma forma que no caso anterior, a extensão do modelo foi avaliada para bases com vários atributos (Tabela 4.2), utilizando-se a mesma configuração definida para o caso de múltiplas classes.

A Tabela 4.51 mostra o comportamento para as bases de múltiplas classes com os métodos de Associação CD e MQR. O objetivo foi testar se as características do CD eram mantidas nesta extensão do modelo AutoFIS. Os resultados mostram que a acurácia, na maioria de bases de dados, foi superior para a associação MQR, além de o desvio padrão ter sido menor. Com respeito ao número de regras, o uso do CD continuou gerando bases de regras menores.

Finalmente, foi avaliada a quantidade de classificadores menores que puderam gerar premissas para as classes: o MQR garantiu uma taxa de sucesso maior do que o CD, como mostrado na Tabela 4.50.

Dataset	CD				MQR			
	25	50	75	100	25	50	75	100
automobile	0	0	3.2	3.1	23.7	45.9	68.3	93.3
penbased	22.4	44.0	66.6	88.5	25	50	75	100
satimage	10.5	20.5	30.6	40.3	25	50	75	100
segment	17.7	35.0	53.0	73.1	25	50	75	100
dermatology	4.0	6.6	10.2	14.3	25	50	75	100
movementlibras	0	0	0	0	25	50	75	100
optdigits	7.8	17.8	25.3	35.9	25	50	75	100
texture	10.7	18.6	32.2	41.7	25	50	75	100

Tabela 4.50: Taxa de sucesso dos classificadores menores nas bases de múltiplas classes.

Dataset	CD				MQR				
	25	50	75	100	25	50	75	100	
Ac	automobile	-	-	64.36	66.48	75.31	75.81	76.30	76.65
	penbased	85.67	85.80	86.04	85.85	85.36	85.75	85.64	85.80
	satimage	80.14	80.09	80.67	80.16	79.86	80.53	80.54	80.62
	segment	86.15	87.14	86.75	87.62	86.37	87.49	86.62	87.88
	dermatology	85.13	87.34	93.30	92.99	96.63	94.96	96.61	96.08
	movementlibras	-	-	-	-	63.06	64.44	65.27	65.00
	optdigits	79.63	86.64	87.99	88.51	85.34	87.76	89.47	90.09
	texture	76.64	77.73	79.29	79.43	81.25	81.38	83.85	83.11
Std _{Ac}	automobile	-	-	5.26	7.00	2.56	3.79	3.59	1.96
	penbased	0.81	0.97	0.85	0.89	1.40	0.89	0.77	0.73
	satimage	1.17	1.00	0.63	1.53	0.93	1.01	0.60	0.73
	segment	2.87	1.47	2.55	1.35	2.47	2.79	2.31	1.67
	dermatology	7.58	6.12	3.34	4.40	2.47	2.76	2.46	2.83
	movementlibras	-	-	-	-	7.46	6.31	7.47	7.26
	optdigits	3.83	1.97	1.61	1.54	1.37	1.46	0.98	1.02
	texture	2.16	2.02	2.08	1.48	1.80	1.45	0.86	1.80
Nr	automobile	-	-	25.5	28.1	41.0	42.1	45.3	43.4
	penbased	101.5	127.3	144.4	150.0	104.7	124.5	140.3	151.1
	satimage	26.2	28.2	31.8	31.1	35.3	39.7	39.9	41.1
	segment	28.8	30.7	33.8	34.3	27.5	28.2	29.0	31.9
	dermatology	48.9	53.8	62.5	62.6	70.5	70.0	74.6	77.9
	movementlibras	-	-	-	-	98.7	98.3	101.2	102.9
	optdigits	110.9	133.1	137.8	148.8	119.0	136.5	145.6	145.8
	texture	29.1	31.1	32.9	32.3	34.4	32.4	31.9	33.9

Tabela 4.51: Extensão do modelo AutoFIS em bases de múltiplas classes.

4.3 Comparações com outros modelos

Esta seção apresenta a comparação do modelo AutoFIS-Class com diversos outros modelos da literatura. Foram utilizados dois conjuntos de dados:

- Grupo 1, do trabalho [35], composto de 26 bases de dados; comparação – acurácia e número de regras – para os seguintes algoritmos evolucionários: 2SLAVE, FH-GBML, SGERD, FARC-HD.
- Grupo 2, do trabalho [33], contendo 19 bases de dados, comparação para os seguintes algoritmos evolucionários: PCA-Ravi, 2SLAVE, GP-PITT-Tsakonas, GCCL-Ishibuchi, FRBCS-GP, GPCOACH. Note-se que são fornecidas as acurácias para os seis modelos, mas o número de regras só é apresentado para três deles.

4.3.1 Resultados e discussões

4.3.1.1 Grupo 1

A Tabela 4.52 apresenta os principais resultados para cada modelo na fase de teste, em termos de acurácia e número de regras, considerando as diferentes bases de dados. A Tabela 4.53 exibe os resultados fornecidos pelo Teste de Friedman, Iman-Davenport e pós-teste de Holm para a acurácia. Observa-se que o algoritmo FARC-HD foi o que obteve o menor posto (1.3846), isto é, obteve uma acurácia tal que o colocou quase sempre em primeiro lugar, ao passo que o AutoFIS-Class ocupou o segundo lugar, com posto igual a 2.7308. Tendo sido o que proporcionou o menor posto, o FARC-HD foi selecionado para ser comparado par a par com os demais algoritmos. Observa-se que o FARC-HD apresentou um posto menor do que os dos demais algoritmos (p -valor $< 0, 05$), significando que a sua acurácia na fase de teste foi significativamente superior às dos demais.

Em termos de compacidade, o AutoFIS-Class gerou menos regras do que o FARC-HD em 73,08% das bases de dados. Ou seja, o objetivo principal de se obter uma base de dados compacta, sem comprometer significativamente a acurácia, foi plenamente atingido. Ressalte-se, ainda, que AutoFIS-Class não executa nenhuma rotina de algoritmo evolutivo (para simplificação ou aprimoramento da base de regras), ao contrário do que é realizado nas demais abordagens.

Base de dados	Acurácia					Número de regras				
	2SLAVE	FH-GBML	SGERD	FARC-HD	AutoFIS	2SLAVE	FH-GBML	SGERD	FARC-HD	AutoFIS
1 Appendicitis	82.91	86.00	84.48	84.18	85.18	4.40	13.80	2.50	6.80	5.40
2 Cleveland	48.82	53.51	51.59	55.24	59.62	11.90	6.90	6.40	61.30	51.30
3 Crx	74.06	86.60	85.03	86.03	85.86	2.40	11.60	2.10	25.40	15.60
4 Ecoli	84.53	69.38	74.05	82.19	75.60	12.60	10.30	9.40	33.80	38.80
5 German	70.53	87.01	67.97	72.80	73.30	6.50	5.10	3.40	85.70	38.30
6 Glass	58.05	57.99	58.49	70.24	55.49	15.10	9.40	6.90	22.70	28.30
7 Heart	71.36	75.93	73.21	84.44	82.22	4.30	12.70	2.70	27.00	17.10
8 Iris	94.44	94.00	94.89	96.00	96.67	4.00	14.90	3.40	4.00	6.90
9 Magic	73.96	81.30	72.06	84.51	79.59	4.10	9.90	3.10	43.30	7.90
10 Monks	97.26	98.18	80.65	99.77	97.27	3.00	14.70	2.20	14.20	9.90
11 Movementlibras	67.04	68.89	68.09	76.67	65.00	47.40	12.10	22.90	83.10	102.90
12 Page-blocks	91.39	94.21	90.72	95.01	93.53	7.50	7.40	6.50	19.10	17.50
13 Pen-based	81.16	50.45	67.93	96.04	85.80	40.00	18.40	15.90	152.80	151.10
14 Phoneme	76.41	79.66	75.55	82.14	72.13	11.50	17.40	3.60	17.80	5.10
15 Pima	73.71	75.26	73.37	75.66	73.84	7.80	10.60	3.10	22.70	7.30
16 Ringnorm	79.63	86.92	72.63	94.08	56.12	4.60	6.90	6.80	24.00	30.40
17 SatImage	81.69	74.72	77.10	87.32	80.62	57.90	16.50	12.20	76.10	41.10
18 Sonar	71.42	68.24	71.90	80.19	79.21	9.60	10.30	3.20	18.00	16.80
19 Spambase	70.14	77.22	72.98	91.93	89.80	7.90	3.90	3.70	29.80	22.80
20 Spectfheart	79.17	72.36	78.16	79.83	79.81	6.10	10.80	2.10	12.90	12.10
21 Texture	81.57	70.15	71.66	92.89	83.11	34.90	14.60	18.60	54.50	33.90
22 Twonorm	86.99	85.97	73.98	95.28	94.53	26.50	12.00	3.10	60.90	31.90
23 Vowel	71.11	67.07	65.83	71.82	49.09	63.10	9.20	18.00	72.30	59.40
24 Wdbc	92.33	92.26	90.68	95.25	94.37	5.20	7.20	3.70	10.40	8.80
25 Wine	89.47	92.61	91.88	94.35	94.90	5.50	9.20	4.20	8.70	9.70
26 Yeast	51.27	51.42	38.77	58.50	51.95	23.60	7.50	11.30	35.20	47.70

Tabela 4.52: Comparação de acurácias e número de regras no Grupo 1.

i	Configuração	Posto
4	SGERD	4.0769
3	2SLAVE	3.6538
2	FH-GBML	3.1538
1	AutoFIS	2.7308
0	FARC-HD	1.3846

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
SGERD	6.1394	0.0001	0.0125	Sim
2SLAVE	5.1746	0.0001	0.0167	Sim
FH-GBML	4.0345	0.0001	0.0250	Sim
AutoFIS	3.0697	0.0021	0.0500	Sim

Tabela 4.53: Resultados do teste de Friedman e Holm para acurácias no Grupo 1.

i	Configuração	Posto
4	FARC-HD	4.5577
3	AutoFIS	3.8462
2	FH-GBML	2.7308
1	2SLAVE	2.6731
0	SGERD	1.1923

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
FARC-HD	7.6743	0.0001	0.0125	Sim
AutoFIS	6.0517	0.0001	0.0167	Sim
FH-GBML	3.5082	0.0005	0.0250	Sim
2SLAVE	3.3767	0.0007	0.0500	Sim

Tabela 4.54: Resultados do teste de Friedman e Holm para o número de regras no Grupo 1.

4.3.1.2 Grupo 2

A Tabela 4.55 apresenta os principais resultados para cada modelo na fase de teste, em termos de acurácia e número de regras, considerando as diferentes bases de dados. Observa-se que o modelo AutoFIS-Class obteve melhores resultados em termos de acurácia. A Tabela 4.56 exhibe os resultados da aplicação do Teste de Friedman, Iman-Davenport e pós-teste de Holm para a acurácia. Observa-se que o algoritmo AutoFIS-Class foi o que obteve menor posto (1.9474), isto é, obteve uma acurácia tal que o colocou quase sempre em primeiro lugar. Tendo sido o que proporcionou o menor posto, o AutoFIS-Class foi selecionado para ser comparado par a par com os demais algoritmos, conseguindo ser significativamente superior (p -valor $< 0,05$) a 4 (GP-PITT-Tsakonas, PCA-Ravi, SLAVE e GCCL-Ishibuchi) dos 6 modelos baseados em Algoritmos Evolucionários. Em termos de compacidade, a Tabela 4.57 exhibe que o AutoFIS-Class obteve o terceiro lugar na geração de regras. Neste grupo de teste o AutoFIS obteve melhores resultados em termos de acurácia e ao mesmo tempo é competitivo no número de regras gerado. Ressalte-se, ainda, que AutoFIS-Class não executa nenhuma rotina de algoritmo evolutivo (para simplificação ou aprimoramento da base de regras), ao contrário do que é realizado nas demais abordagens.

	Base de dados	Acurácia							Número de regras			
		PCA.Ravi	2SLAVE	GP.PITT	GCCL	FRBCS-GP	GPCOACH	AutoFIS	2SLAVE	FRBCS-GP	GP-COACH	AutoFIS
1	Bupa	54.46	58.58	56.45	58.27	62.20	63.63	60.08	4.57	18.53	10.07	4.30
2	Cleveland	49.24	46.19	56.46	54.15	56.69	55.23	59.62	12.37	34.53	23.83	51.30
3	Ecoli	55.46	57.49	43.94	71.17	76.75	77.72	75.60	10.37	30.40	25.57	38.80
4	Glass	46.56	44.39	45.12	60.69	56.61	65.33	55.49	8.80	23.47	17.43	28.30
5	Iris	88.44	94.67	48.44	94.67	97.11	97.56	96.67	3.93	3.00	3.23	6.90
6	Libras Mov.	42.41	25.83	5.28	20.74	47.69	45.56	65.00	25.53	49.77	113.93	102.90
7	Magic	77.59	74.29	64.79	76.02	74.51	79.82	79.59	4.23	33.33	9.33	7.90
8	Page-blocks	90.70	91.42	92.92	90.34	91.09	91.23	93.53	7.53	39.87	14.97	17.50
9	Pen-based	81.81	81.16	44.67	82.18	75.53	82.20	85.80	39.97	87.07	89.70	151.10
10	Pima	68.31	66.45	64.28	69.11	73.16	74.37	73.84	3.80	27.90	17.23	7.30
11	Ring	30.12	79.64	50.51	91.70	93.84	91.13	56.12	4.60	39.60	17.50	30.40
12	Satimage	76.54	33.45	23.82	63.12	68.06	72.50	80.62	9.83	93.40	27.53	41.10
13	Segment	78.50	72.81	21.62	84.07	80.38	85.96	87.88	10.47	38.80	23.30	31.90
14	Sonar	27.65	70.72	52.42	72.40	71.15	67.48	79.21	9.33	20.97	14.03	16.80
15	Spambase	64.93	70.14	81.89	69.87	74.55	82.80	89.80	7.90	43.93	10.27	22.80
16	Twonorm	20.04	84.35	48.80	90.12	91.97	84.83	94.53	24.40	99.27	51.67	31.90
17	Wdbc	86.77	91.80	63.09	91.09	95.02	93.90	94.37	5.47	16.30	4.90	8.80
18	Wine	93.17	91.53	38.19	91.21	91.13	95.10	94.90	5.73	9.60	7.57	9.70
19	Yeast	39.45	14.51	31.76	49.01	52.16	48.56	51.95	13.27	64.03	32.20	47.70

Tabela 4.55: Comparação de acurácias e número de regras no Grupo 2.

i	Configuração	Posto
6	GP-PITT-Tsakonas	6.0000
5	PCA-Ravi	5.3158
4	2SLAVE	5.1316
3	GCCL-Ishibuchi	4.1316
2	FRBCS-GP	3.0526
1	GPCOACH	2.4211
0	AutoFIS	1.9474

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
GP-PITT-Tsakonas	5.7822	0.0001	0.0083	Sim
PCA-Ravi	4.8060	0.0001	0.0100	Sim
2SLAVE	4.5432	0.0001	0.0125	Sim
GCCL-Ishibuchi	3.1164	0.0018	0.0167	Sim
FRBCS-GP	1.5770	0.1148	0.0250	Não
GPCOACH	0.6758	0.4991	0.0500	Não

Tabela 4.56: Resultados do teste de Friedman e Holm para a acurácia no Grupo 2.

i	Configuração	Posto
3	FRBCS-GP	3.4211
2	AutoFIS	3.0526
1	GP-COACH	2.3158
0	2SLAVE	1.2105

Teste	p-valor
Friedman	<0.0001
Iman e Davepor	<0.0001

Configuração	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
FRBCS-GP	5.2776	0.0001	0.0083	Sim
AutoFIS	4.3980	0.0001	0.0250	Sim
GP-COACH	2.6388	0.0083	0.0500	Não

Tabela 4.57: Resultados do teste de Friedman e Holm para o número de regras no Grupo 2.

5

Conclusões e Trabalhos Futuros

Esta dissertação abordou o desenvolvimento do modelo Automatic Fuzzy Inference System for Classification problems (AutoFIS-Class), uma nova metodologia para a geração automática de Sistemas de Inferência Fuzzy em Classificação. Foram exibidas suas principais características, tais como (i) gerar bases de regras competitivas sem uso de um Algoritmo Evolucionário; (ii) possibilitar o uso de operadores de agregação e negação; (iii) empregar heurísticas de definição do consequente mais apropriado para um determinado antecedente.

Durante as investigações empíricas, pode-se analisar a influência de cada parâmetro do modelo na acurácia e na compacidade da base de regras. Os resultados mostraram-se superiores, aos de alguns modelos de evolucionários.

Os experimentos realizados tiveram como objetivo ilustrar o funcionamento do modelo e evidenciar a sua qualidade. O modelo AutoFIS-Class abre diversas frentes para trabalhos futuros. A seguir são sugeridas algumas linhas, de acordo com a arquitetura do modelo: Fuzzificação, Associação, Agregação e Avaliação.

- **Funções de Pertinência:** os formatos triangular fortemente particionado e Tukey demonstraram ser úteis; ambos geraram bons resultados nas bases de dados analisadas. Entretanto, deve-se buscar um indicador de forma a se escolher, automaticamente, o formato dos conjuntos, já que o formato Tukey gera uma elevação do tempo computacional.
- **Granularidade:** nos experimentos realizados, o uso de três ou cinco funções de pertinência apresentou grande influência na acurácia de certas bases de dados. Assim, o desenvolvimento de uma metodologia que determine um particionamento automático entre estes dois valores incrementaria o desempenho do modelo.
- **Operadores Fuzzy:** ao longo das aplicações apresentadas, o conjunto de operadores fuzzy ficou restrito a mínimo, produto e negação clássica. Entretanto, é sempre possível investigar o efeito de outras t-normas, modificadores linguísticos e negações. Embora haja algum prejuízo em termos de interpretabilidade, t-conormas também podem ser empregadas.
- **Tamanho das premissas:** embora a negação tenha proporcionado melhor acurácia, o custo computacional envolvido não é trivial. Sua

utilização deve ser cuidadosamente avaliada em premissas de tamanho superior a dois.

- **Operadores de Agregação:** dos operadores investigados, os baseados em Mínimos Quadrados Restritos (MQR e IntMQR) apresentaram melhor desempenho em termos de acurácia. O MQR apresentou o melhor compromisso entre acurácia e compacidade da base de regras fuzzy. Porém, houve reflexo na elevação do tempo computacional. Há a possibilidade de se considerarem outros operadores como os do tipo não-linear.
- **Métodos de Associação:** o método CD foi adequado para as bases binárias mas não para as múltiplas classes, em que os melhores resultados foram obtidos com uso do MQR – embora com eventual prejuízo em termos de interpretabilidade. Nesse sentido, é possível investir em novas heurísticas que definam o consequente mais apropriado, sobretudo nas bases de múltiplas classes.

Aproveitando a natureza determinística do modelo, a extensão proposta para uso com bases de dados com alta dimensionalidade em número de atributos pode ser aprimorada por meio da introdução de aleatoriedade nos parâmetros dos classificadores menores. Nesta dissertação foi desenvolvido um *bagging* [47] de classificadores menores com a mesma arquitetura, cujas premissas geradas foram agregadas na etapa final. Os resultados foram aceitáveis e abre-se um caminho de construção e avaliação de classificadores com diversas arquiteturas ao estilo do *Random Forest*. A grande vantagem desta abordagem é a redução do custo computacional na geração de classificadores, além de relaxar a dependência da configuração dos parâmetros do modelo. Além disso, existe a oportunidade de paralelização do algoritmo, o que reduziria o tempo de execução.

Finalmente, a estrutura do modelo AutoFIS-Class – com a definição da Defuzzificação – pode ser estendida para outros problemas, como, por exemplo, de Regressão.

Referências Bibliográficas

- [1] BAITHARU, T.; PANI, S.. **A Survey on Application of Machine Learning Algorithms on Data Mining**. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 3(7):17–20, 2013.
- [2] PHYU, T.. **Survey of Classification Techniques in Data Mining**. International Multiconference of Engineers and Computer Scientists, 1:18–20, 2009.
- [3] ALONSO, J.; MAGDALENA, L. ; GONZÁLEZ-RODRÍGUEZ, G.. **Looking for a good fuzzy system interpretability index: An experimental approach**. International Journal of Approximate Reasoning, 51(1):115–134, 2009.
- [4] MARTENS, D.; BAESENS, B. ; VAN GESTEL, T.. **Decompositional Rule Extraction from Support Vector Machines by Active Learning**. IEEE Transactions on Knowledge and Data Engineering, 21(2):178–191, 2009.
- [5] TSUKIMOTO, H.. **Extracting rules from trained neural networks**. IEEE Transactions on Neural Networks, 11(2):377–389, 2000.
- [6] LUGHOFER, E.. **On-line assurance of interpretability criteria in evolving fuzzy systems - Achievements, new concepts and open issues**. Information Sciences, 251:22–46, 2013.
- [7] CORDÓN, O.. **A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems**. International Journal of Approximate Reasoning, 52(6):894–913, 2011.
- [8] GACTO, M. J.; ALCALÁ, R. ; HERRERA, F.. **Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures**. Information Sciences, 181(20):4340–4360, 2011.
- [9] FAZZOLARI, M.; ALCALA, R.; NOJIMA, Y.; ISHIBUCHI, H. ; HERRERA, F.. **A Review of the Application of Multiobjective Evolutionary**

- Fuzzy Systems : Current Status and Further Directions. IEEE Transactions on Fuzzy Systems, 21(1):45–65, 2013.
- [10] ISHIBUCHI, H.; YAMANE, M. ; NOJIMA, Y.. Rule weight update in parallel distributed fuzzy genetics-based machine learning with data rotation. IEEE International Conference on Fuzzy Systems, 2013.
- [11] HERRERA, F.. Genetic fuzzy systems: Taxonomy, current research trends and prospects. Evolutionary Intelligence, 1(1):27–46, 2008.
- [12] LEITE, D.; BALLINI, R.; COSTA, P. ; GOMIDE, F.. Evolving fuzzy granular modeling from nonstationary fuzzy data streams. Evolving Systems, 3(2):65–79, 2012.
- [13] ANGELOV, P.; ZHOU, X.. Evolving fuzzy-rule-based classifiers from data streams. IEEE Transactions on Fuzzy Systems, 16(6):1462–1475, 2008.
- [14] FERNÁNDEZ, A.; LÓPEZ, V.; DEL JESUS, M. ; HERRERA, F.. Revisiting Evolutionary Fuzzy Systems: Taxonomy, applications, new trends and challenges. Knowledge-Based Systems, 80:109–121, 2015.
- [15] KARR, C.. Genetic Algorithms for Fuzzy Controllers. AI Expert, 6:26–33, 1991.
- [16] VALENZUELA RENDÓN, M.. The fuzzy classifier system: A classifier system for continuously varying variables. In: PROCEEDINGS OF THE 4TH INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, p. 223–230. Morgan Kaufmann, 1991.
- [17] THRIFT, P.. Fuzzy logic synthesis with genetic algorithms. In: PROCEEDINGS OF THE 4TH INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, p. 514–518. Morgan Kaufmann, 1991.
- [18] KOSHIYAMA, A.. GPFIS : Um Sistema Fuzzy-Genético Genérico baseado em Programação Genética. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro, 2014.
- [19] ISHIBUCHI, H.; NOZAKI, K.; YAMAMOTO, N. ; TANAKA, H.. Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Transactions on Fuzzy Systems, 3(3):260–270, 1995.
- [20] ISHIBUCHI, H.; MURATA, T. ; TURKSEN, I.. Single-objective and two-objective genetic algorithms for selecting linguistic rules for

- pattern classification problems. *Fuzzy Sets and Systems*, 89(2):135–150, July 1997.
- [21] CASILLAS, J.; CARSE, B. ; BULL, L.. **Fuzzy-XCS: A Michigan genetic fuzzy system.** *IEEE Transactions on Fuzzy Systems*, 15(4):536–550, 2007.
- [22] CORDON, O.; DEL JESUS, M.; HERRERA, F. ; LOZANO, M.. **MOGUL: A Methodology to Obtain Genetic Fuzzy Rule-Based Systems under the Iterative Rule Learning Approach.** *International Journal of Intelligent Systems*, 14:1123–1153, 1999.
- [23] GONZÁLEZ, A.; PÉREZ, R.. **SLAVE: A genetic learning system based on an iterative approach.** *IEEE Transactions on Fuzzy Systems*, 7(2):176–191, 1999.
- [24] GONZÁLEZ, A.; PÉREZ, R.. **Selection of relevant features in a fuzzy genetic learning algorithm.** *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 31(3):417–425, 2001.
- [25] GARCÍA, D.; GONZÁLEZ, A. ; PÉREZ, R.. **A feature construction approach for genetic iterative rule learning algorithm.** *Journal of Computer and System Sciences*, 80(1):101–117, 2014.
- [26] HO, S.; CHEN, H.; HO, S. ; CHEN, T.. **Design of Accurate Classifiers With a Compact Fuzzy-Rule Base Using an Evolutionary Scatter Partition of Feature Space.** *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(2):1031–1044, 2004.
- [27] MUCIENTES, M.; VIDAL, J.; BUGARÍN, A. ; LAMA, M.. **Processing time estimations by variable structure TSK rules learned through genetic programming.** *Soft Computing*, 13(5):497–509, 2009.
- [28] REGATTIERI DELGADO, M.; VON ZUBEN, F. ; GOMIDE, F.. **Hierarchical genetic fuzzy systems.** *Information Sciences*, 136(1-4):29–52, Aug. 2001.
- [29] ISHIBUCHI, H.; YAMAMOTO, T. ; NAKASHIMA, T.. **Hybridization of Fuzzy GBML Approaches for Pattern Classification Problems.** *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 35(2):359–365, Apr. 2005.
- [30] FERNÁNDEZ, A.; CALDERÓN, M.; BARRENECHEA, E.; BUSTINCE, H. ; HERRERA, F.. **Solving multi-class problems with linguistic fuzzy**

- rule based classification systems based on pairwise learning and preference relations. *Fuzzy Sets and Systems*, 161(23):3064–3080, 2010.
- [31] SANZ, J.; FERNANDEZ, A.; BUSTINCE, H. ; HERRERA, F.. **IVTURS: A linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection.** *IEEE Transactions on Fuzzy Systems*, 21(3):399–411, 2013.
- [32] TSAKONAS, A.. **A comparison of classification accuracy of four genetic programming-evolved intelligent structures.** *Information Sciences*, 176(6):691–724, 2006.
- [33] BERLANGA, F. J.; RIVERA, A. J.; DEL JESUS, M. J. ; HERRERA, F.. **GP-COACH: Genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems.** *Information Sciences*, 180(8):1183–1200, 2010.
- [34] LÓPEZ, V.; FERNÁNDEZ, A.; DEL JESUS, M. ; HERRERA, F.. **A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets.** *Knowledge-Based Systems*, 38:85–104, 2013.
- [35] ALCALÁ-FDEZ, J.; ALCALÁ, R. ; HERRERA, F.. **A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning.** *IEEE Transactions on Fuzzy Systems*, 19(5):857–872, 2011.
- [36] MUNI, D.; PAL, N.. **Evolution of fuzzy classifiers using genetic programming.** *Fuzzy Information and Engineering*, 4(1):29–49, 2012.
- [37] KOSHIYAMA, A.; ESCOVEDO, T.; DIAS, D.; VELLASCO, M. ; TANSCHKEIT, R.. **GPF-CLASS: A Genetic Fuzzy model for classification.** In: 2013 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, p. 3275–3282. IEEE, June 2013.
- [38] LEMOS, A.; CAMINHAS, W. ; GOMIDE, F.. **Evolving Intelligent Systems: Methods, Algorithms and Applications.** *Emerging Paradigms in Machine Learning*, 13:117–159, 2013.
- [39] MENCAR, C.; FANELLI, A. M.. **Interpretability constraints for fuzzy information granulation.** *Information Sciences*, 178(24):4585–4618, 2008.

- [40] ZHOU, S.; GAN, J.. **Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling.** *Fuzzy Sets and Systems*, 159(23):3091–3131, 2008.
- [41] PAP, E.; KLEMENT, E. ; MESIAR, R.. **Triangular norms.** Kluwer Academic Publishers, 2000.
- [42] Calvo, T.; Mayor, G. ; Mesiar, R., editors. **Aggregation Operators**, volumen 97. Physica-Verlag Heidelberg, 2002.
- [43] CALVO, T.; KOLESÁROVÁ, A.; KOMORNÍKOVÁ, M. ; MESIAR, R.. **Aggregation Operators: Properties, Classes and Construction Methods.** *Aggregation Operators: New Trends and Applications*, 97(1):3–104, 2002.
- [44] DERRAC, J.; GARCÍA, S.; MOLINA, D. ; HERRERA, F.. **A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms.** *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.
- [45] ALCALÁ-FDEZ, J.; FERNÁNDEZ, A.; LUENGO, J.; DERRAC, J.; GARCÍA, S.; SÁNCHEZ, L. ; HERRERA, F.. **KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework.** *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.
- [46] BACHE, K.; LICHMAN, M.. **UCI machine learning repository**, 2013.
- [47] GALAR, M.; FERNANDEZ, A.; BARRENECHEA, E.; BUSTINCE, H. ; HERRERA, F.. **A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches.** *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 42(4):463–484, 2012.