

Síntese Automática de Sistemas de Inferência Fuzzy para Classificação

Jorge Paredes, Adriano S. Koshiyama, Marley M.B.R. Vellasco e Ricardo Tanscheit
Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225 – Gávea/Rio de Janeiro, RJ
Email: [jparedes, adriano, marley, ricardo]@ele.puc-rio.br

Resumo—Este trabalho introduz o modelo AutoFIS-Class, destinado à geração automática de Sistemas de Inferência Fuzzy para problemas de Classificação. Esta abordagem consiste em cinco etapas: (i) mapeamento de cada padrão a um grau de pertinência a conjuntos fuzzy; (ii) geração de um conjunto de premissas de regras fuzzy, inspiradas em uma árvore de busca, e aplicação de critérios de qualidade para reduzir o crescimento exponencial; (iii) associação de uma determinada premissa a um conseqüente adequado; (iv) agregação das regras fuzzy para a uma mesma classe e (v) decisão sobre qual classe é mais compatível com um determinado padrão. Na avaliação do AutoFIS-Class, foram utilizadas 21 bases de dados e os resultados foram comparados com os de outros quatro sistemas baseados em regras. Os resultados mostram que o AutoFIS-Class é competitivo frente a estes sistemas, em sua maioria do tipo evolucionário.

Keywords—Sistema de Inferência Fuzzy, Síntese Automática, Classificação.

I. INTRODUÇÃO

Hoje em dia, grande parte do conhecimento acumulado está armazenado em forma de dados. Muitos algoritmos para tarefas de classificação têm sido desenvolvidos para a extração deste conhecimento [5], [21]. Dentre as ferramentas capazes de atuar como modelos representativos de sistemas reais, os Sistemas de Inferência Fuzzy [3] são consagrados no que diz respeito à representação de conhecimento de forma humanamente compreensível por meio de regras de inferência.

A questão da representação adequada do conhecimento é relevante pois em várias aplicações não se deseja apenas um modelo do tipo “caixa preta” [20], que, por mais precisão que proporcione, não fornece uma explicação lógica, funcional ou descritiva de como os resultados são obtidos. Em outras palavras, além de acurácia, busca-se também interpretabilidade linguística, cuja medição não é trivial. Sistemas de Inferência Fuzzy [14], [9] surgem como uma alternativa de modelagem que une capacidade de interpretação a nível linguístico com acurácia satisfatória. Em Sistema de Inferência Fuzzy a avaliação desta interpretabilidade leva em conta, por exemplo, a estrutura do modelo (Mamdani, Takagi-Sugeno, etc.), o número de regras, o número de elementos antecedentes e o formato das funções de pertinência. [9], [13].

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 [16], [14], [11], ou (ii) uso de Sistemas de Inferência Fuzzy Evolutivos [19], [4], [20], 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 custo computacional e o número de parâmetros adicionados em decorrência do uso de um algoritmo evolucionário – em alguns casos multi-objetivo –, ao passo que a abordagem (ii), ao não presumir uma alocação inicial e fixa para as funções de pertinência, torna as regras geradas dificilmente interpretáveis.

Este trabalho busca um meio termo, isto é, uma abordagem que gere um Sistema de Inferência Fuzzy de forma automática, que busque acurácia, mas privilegiando a geração de uma base de regras que valorize a acepção linguística. Propõe-se, portanto, o modelo AutoFIS-Class, um método 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 os graus de ativação de cada regra para uma mesma classe, a partir de operadores de agregação que ponderem cada regra com base em sua influência na classificação. Nos estudos de casos, tomando por base o trabalho de Alcalá-Fdez et al. [2] e considerando 21 bases de dados, compara-se o desempenho do AutoFIS-Class com os de outros quatro Sistemas Fuzzy Evolucionários para geração de regras.

A próxima seção apresenta o modelo AutoFIS-Class, discorrendo sobre as principais etapas da síntese automática de um Sistema de Inferência Fuzzy. A terceira seção descreve os experimentos realizados e apresenta os resultados e discussões. Por fim, a quarta seção conclui o trabalho e sugere trabalhos futuros.

II. MODELO AUTOFIS-CLASS

Descrevem-se, nesta seção, as etapas para elaboração do modelo AutoFIS-CLASS, a começar pelo mapeamento de valores precisos em graus de pertinência a conjuntos fuzzy (Fuzzificação). O processo de Inferência compreende a elaboração das premissas de regras (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 do modelo são mostradas na Figura 1.

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, separam-se as premissas para

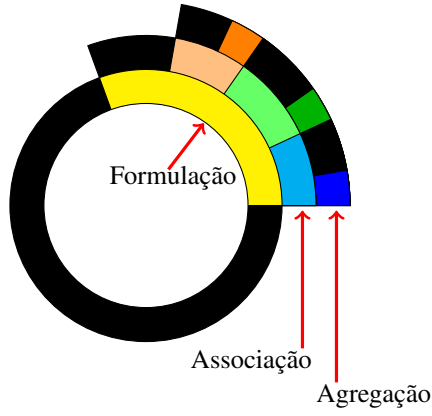


Figura 1. Etapa de Inferência no modelo AutoFIS-Class.

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.

A. Fuzzificação

Em classificação, a principal informação disponível consiste nos n padrões $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iJ}]$ dos J atributos X_j presentes na base de dados ($i = 1, \dots, n$ e $j = 1, \dots, J$). 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 $\mu_{A_{jl}}(x_{ij})$ ao conjunto fuzzy A_{jl} . Por fim, cada padrão está associado a uma classe C_i das K possíveis, isto é, $C_i \in \{1, 2, \dots, k, \dots, K\}$.

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, ambas as 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 [6], [16], [18] empregar uma disposição dita Fortemente Particionada, conforme mostrado na Figura 2a.

Uma outra abordagem faz uso da informação dos quartis: 0º quartil (valor 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 2b. Ressalte-se que, neste trabalho, os atributos categóricos e as classes são binarizadas.

B. Inferência Fuzzy

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

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

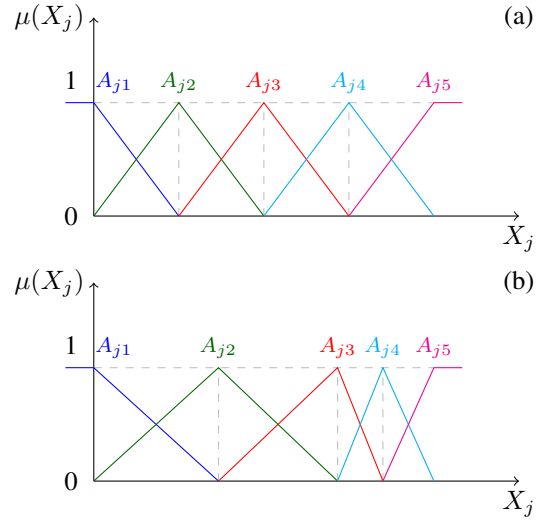


Figura 2. Funções de pertinência triangulares: (a) Fortemente Particionada e (b) Tukey.

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 a partir de uma t-norma $*$ que combina cada $\mu_{A_{jl}}(x_{ij})$ marginal. Assim, 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 \dots e X_j Não é A_{jl} e \dots e X_J Não é A_{Jl} ”. Este procedimento, em oposição a negar o resultado como um todo [Não(“Se X_1 é A_{1l} e \dots e X_j é A_{jl} e \dots e X_J é A_{Jl})”], dobra o número de possibilidades de premissas, ampliando ainda mais o espaço de busca.

Apesar do emprego restrito da t-norma produto e do operador de negação por elemento, é sabido que 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). 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} ”; 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 em consequência uma quantidade imensa de premissas. Contudo, nem todas as premissas serão usadas para formar 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:

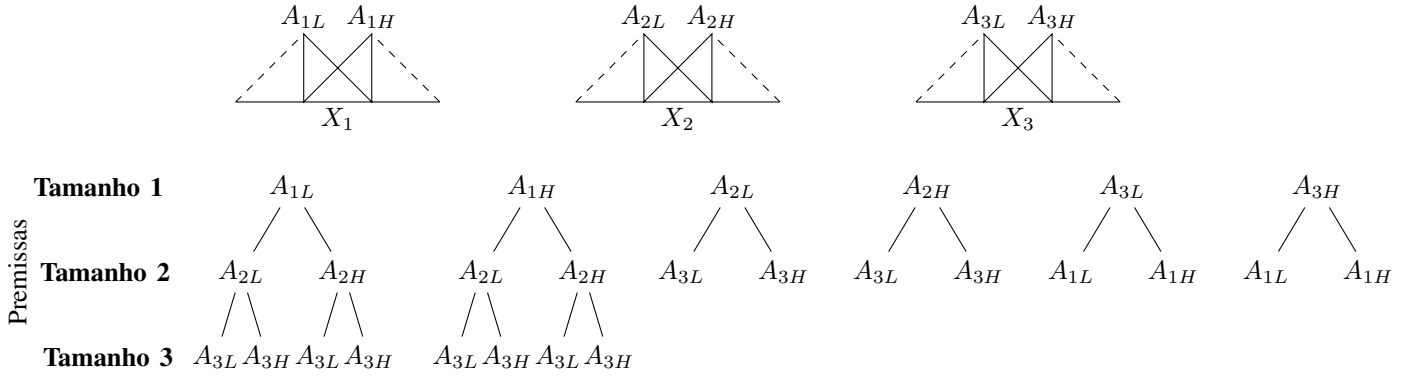


Figura 3. Premissas agrupadas pelo tamanho das premissas para 3 atributos com 2 conjuntos fuzzy cada um.

- 1) Limitação no tamanho máximo das 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 *menor para o maior não exaustivo*, isto é, a inicialização das premissas pelas 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).

a) 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:

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

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 [17]. Tal observação é ilustrada nas Figuras 4 e 5.

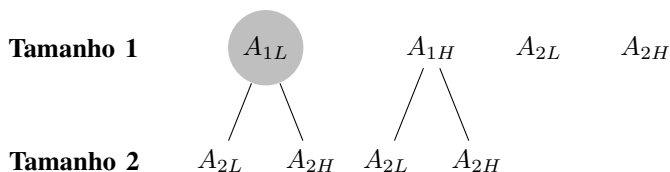


Figura 4. Premissa que não excede o limiar do Suporte

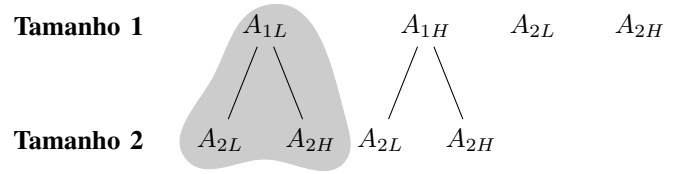


Figura 5. Premissas dispensadas do espaço de busca

b) Filtro da 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:

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

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 ainda assim excluí-la do processo de forma a gerar o menor número possível de premissas para os demais passos pós-formulação.

c) Filtro do PCD: Este filtro busca reduzir a ocorrência de regras semelhantes ou conflitantes. Para tanto, calcula-se o Grau de Confiança Penalizado [12] - PCD_k -, dado por:

$$PCD_k = \max \left\{ \frac{\sum_{i \in k} \mu_{A_d}(\mathbf{x}_i) - \sum_{i \notin k} \mu_{A_d}(\mathbf{x}_i)}{\sum_{i=1}^n \mu_{A_d}(\mathbf{x}_i)}, 0 \right\} \quad (4)$$

O objetivo do PCD_k é 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. Uma premissa não é viável se $PCD_k = 0$ para todas as K classes.

Após a geração de D premissas $\mu_{A_1}(\mathbf{x}_i), \mu_{A_2}(\mathbf{x}_i), \dots, \mu_{A_D}(\mathbf{x}_i)$ que atendam aos limites de tamanho e que passem por todos os filtros, o próximo passo é associar cada premissa a uma classe consequente.

B.2) *Associação*: A etapa de Associação determina 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 diversas métricas possíveis, o AutoFIS emprega a o Grau de Confiança (CD_k) [15], [6]:

$$CD_k = \frac{\sum_{i \in k} \mu_{A_d}(\mathbf{x}_i)}{\sum_{i=1}^n \mu_{A_d}(\mathbf{x}_i)} \in [0, 1] \quad (5)$$

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 direcionada a classe consequente alguma.

B.3) *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 reúne o grau de ativação de \mathbf{x}_i^* para as 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 [7], $g : [0, 1]^{D^{(k)}} \rightarrow [0, 1]$, o grau de pertinência predito de \mathbf{x}_i^* a cada uma das K classes ($\hat{\mu}_{C_i \in k}(\mathbf{x}_i^*)$) é:

$$\hat{\mu}_{C_i \in 1}(\mathbf{x}_i^*) = g[\mu_{A_{1(1)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(1)}}(\mathbf{x}_i^*)] \quad (6)$$

$$\hat{\mu}_{C_i \in 2}(\mathbf{x}_i^*) = g[\mu_{A_{1(2)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(2)}}(\mathbf{x}_i^*)] \quad (7)$$

$$\dots$$

$$\hat{\mu}_{C_i \in K}(\mathbf{x}_i^*) = g[\mu_{A_{1(K)}}(\mathbf{x}_i^*), \dots, \mu_{A_{D(K)}}(\mathbf{x}_i^*)] \quad (8)$$

A t-conorma mais usada como operador de agregação em Sistemas de Inferência Fuzzy é o máximo. Uma alternativa é o uso de uma média ponderada, similarmente ao procedimento AsMQR [7]. Propõe-se, assim, o uso da Média Ponderada estimada via Mínimos Quadrados Restritos (MPMQR):

$$\hat{\mu}_{C_i \in 1}(\mathbf{x}_i^*) = \sum_{d(1)=1}^{D(1)} w_{d(1)} \mu_{A_{d(1)}}(\mathbf{x}_i^*) \quad (9)$$

$$\dots$$

$$\hat{\mu}_{C_i \in K}(\mathbf{x}_i^*) = \sum_{d(K)=1}^{D(K)} w_{d(K)} \mu_{A_{d(K)}}(\mathbf{x}_i^*) \quad (10)$$

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 .

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

C. Decisão

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

$$\hat{C}_i = \arg_k \max \{ \hat{\mu}_{C_i \in 1}(\mathbf{x}_i^*), \dots, \hat{\mu}_{C_i \in k}(\mathbf{x}_i^*), \dots, \hat{\mu}_{C_i \in K}(\mathbf{x}_i^*) \} \quad (11)$$

onde \hat{C}_i é a classe predita: resultado do k -ésimo argumento que maximiza a expressão (11). Assim, esse método associa ao padrão \mathbf{x}_i^* à classe em que 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 a \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):

$$EMC = \frac{\sum_{i=1}^n |C_i - \hat{C}_i|}{n} \quad (12)$$

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

III. ESTUDOS DE CASOS

A. Descrição dos Experimentos

As bases de dados usadas nos experimentos – 21 das 26 empregadas por Alcalá-Fdez et al. [2] – foram obtidas no site da Ferramenta KEEL [1] e são apresentadas na Tabela I.

Tabela I. PRINCIPAIS CARACTERÍSTICAS DAS BASES DE DADOS USADAS.

Base	J	n	K	Base	J	n	K
Iris	4	150	3	Cleveland	13	297	5
Phoneme	5	5404	2	Vowel	13	990	11
Monk-2	6	432	2	Crx	15	653	2
Appendicitis	7	106	2	Pen-based	16	10992	10
Pima	8	768	2	Two-norm	20	7400	2
Glass	9	214	6	Wdbc	30	569	2
Page-blocks	10	5472	5	Satimage	36	6435	6
Magic	10	19020	2	Texture	40	5500	11
Wine	13	178	3	German	20	1000	2
Heart	13	270	2	Spambase	57	4597	2
Ring	20	7400	2				

Observa-se que há bases de dados de tamanho pequeno (Iris, Appendicitis e Wine) e outras de elevada escalabilidade e dimensionalidade (Spambase, Texture, Satimage, etc.). No total, foram usadas 11 bases de dados binárias e 10 de classes múltiplas. O trabalho realizado em [2] traz resultados para quatro modelos: 2SLAVE, FH-GBML, SGERD e FARC-HD, todos eles do tipo Fuzzy-Evolutivos, com o objetivo de gerar bases de regras fuzzy concisas a partir do uso de uma meta-heurística evolutiva. Deve-se destacar o modelo FARC-HD (Fuzzy Association Rule-Based Classification Model for High-Dimensional), abordagem proposta em [2], em que um algoritmo evolutivo seleciona um subconjunto das regras inicialmente geradas por meio de um procedimento similar ao do AutoFIS-Class, mas sem usar alguns dos filtros.

Os experimentos com o AutoFIS-Class seguiram o mesmo procedimento de [2]: (i) para cada base de dados efetua-se um processo de validação cruzada de 10 pastas (10-fold-cv); (ii) para cada pasta da validação cruzada, executa-se o AutoFIS-Class, obtendo-se as métricas de acurácia na fase de teste,

assim como o número de regras obtidas; e (iii) a média das métricas nas 10 pastas é a reportada como resultado.

A Tabela II mostra as configurações usadas nos experimentos com o AutoFIS-Class para as bases binárias e de múltiplas classes. Tais configurações foram obtidas de testes preliminares destinados a avaliar (de forma superficial) a sensibilidade do modelo aos diversos parâmetros. Observe-se que as principais diferenças entre as configurações residem no aumento do número de funções de pertinência e na desabilitação do filtro PCD para as bases de dados de múltiplas classes. Este filtro tende a eliminar premissas compatíveis com mais de uma classe, o que pode ser indesejável, pois as poucas regras geradas passam a ser desconsideradas. As análises estatísticas seguiram as recomendações de [8], [10]; adotou-se um nível de significância de 5%.

B. Resultados e Discussões

A Tabela III 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 FARC-HD obteve os melhores resultados em termos de acurácia, superiores aos proporcionados pelo AutoFIS-Class em torno de 4% em média.

Para verificar se a diferença de acurácia entre os modelos é significativa, a Tabela IV apresenta os resultados da aplicação do Teste de Friedman, Iman-Davenport e pós-teste de Holm [10]. Observa-se que o algoritmo FARC-HD foi o que obteve menor posto (1.3333), 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 2.6190. 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 76,19% 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.

Tabela II. CONFIGURAÇÕES DO AUTOFIS-CLASS PARA O TIPO DE BASE DE DADOS

Configuração	Bases Binárias	Bases Múltiplas
Formato da função de pertinência	Tukey	Tukey
Nº funções de pertinência	3	5
t-norma	prod	prod
Negação	Sim	Sim
Tamanho da premissa	2	2
ε_{sup}	0,05	0,05
ε_{sim}	0,95	0,95
Filtro PCD	Sim	Não
Associação	CD	CD
Agregação	MQR	MQR

Tabela IV. RESULTADOS DO TESTE DE FRIEDMAN E HOLM PARA A COMPARAÇÃO ENTRE OS MODELOS.

i	Método	Posto		
4	SGERD	4.1905		
3	2SLAVE	3.7619		
2	FH-GBML	3.0952		
1	AutoFIS-Class	2.6190		
0	FARC-HD	1.3333		
<hr/>				
Teste	p-valor			
Friedman	< 0.0001			
Iman e Daveport	< 0.0001			
<hr/>				
Método	$z = (R_0 - R_i)/SE$	p-valor	Holm	Rejeita?
SGERD	5.8551	<0.0001	0.0125	Sim
2SLAVE	4.9770	<0.0001	0.0167	Sim
FH-GBML	3.6108	0.0003	0.0250	Sim
AutoFIS-Class	2.6349	0.008415	0.0500	Sim

IV. CONCLUSÕES

Este trabalho apresentou uma nova metodologia para a geração automática de Sistemas de Inferência Fuzzy para Classificação, denominada AutoFIS-Class (Automatic Fuzzy Inference System for Classification Problems).

Nos experimentos, foram usadas 21 bases de dados, para as quais o AutoFIS-Class, apesar de não usar nenhum algoritmo evolutivo, obteve o segundo melhor resultado. Além disso, cumpriu o objetivo de gerar uma base de regras compacta.

Trabalhos futuros deverão abordar uma hibridização com o algoritmo de Wang & Mendel [22] para geração de premissas de maior tamanho, o emprego de métodos de agregação não-lineares, novos procedimentos para associação e novos filtros. Aplicações práticas também estão previstas, dada a facilidade de execução da abordagem proposta.

REFERÊNCIAS

- [1] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. 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):255–287, 2011.
- [2] Jesús Alcalá-Fdez, Rafael Alcalá, and Francisco Herrera. 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.
- [3] José M. Alonso, Luis Magdalena, and Gil González-Rodríguez. Looking for a good fuzzy system interpretability index: An experimental approach. *International Journal of Approximate Reasoning*, 51(1):115–134, 2009.
- [4] Plamen P. Angelov and Xiaowei Zhou. Evolving fuzzy-rule-based classifiers from data streams. *IEEE Transactions on Fuzzy Systems*, 16(6):1462–1475, 2008.
- [5] Tapas Ranjan Baitharu and Subhendu Kumar Pani. A Survey on Application of Machine Learning Algorithms on Data Mining. (7):17–20, 2013.
- [6] F.J. Berlanga, A.J. Rivera, M.J. del Jesus, and F. Herrera. 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.
- [7] Tomasa Calvo, Anna Kolesárová, Magda Komorníková, and Radko Mesiar. Aggregation operators: Properties, classes and construction methods. In Tomasa Calvo, Gaspar Mayor, and Radko Mesiar, editors, *Aggregation Operators*, volume 97 of *Studies in Fuzziness and Soft Computing*, pages 3–104. Physica-Verlag HD, 2002.
- [8] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley, New York, 2 edition, 1980.

Tabela III. ACURÁCIA NA FASE DE TESTE E NÚMERO DE REGRAS (# R) OBTIDO PARA CADA ABORDAGEM POR BASE DE DADOS USADA.

Base de Dados	AutoFIS-Class		2SLAVE		FH-GBML		SGERD		FARC-HD	
	Tst. (%)	# R	Tst. (%)	# R	Tst. (%)	# R	Tst. (%)	# R	Tst. (%)	# R
Appendicitis	89,82	5,6	82,91	4,4	86,00	13,8	84,48	2,5	84,18	6,8
Crx	86,77	19,1	74,06	2,4	86,60	11,6	85,03	2,1	86,03	25,4
German	71,40	26,8	70,53	6,5	87,01	5,1	67,97	3,4	72,8	85,7
Heart	83,33	23,6	71,36	4,3	75,93	12,7	73,21	2,7	84,44	27,0
Magic	79,91	7,7	73,96	4,1	81,23	9,9	72,06	3,1	84,51	43,3
Monk-2	97,57	9,7	97,26	3,0	98,18	14,7	80,65	2,2	99,77	14,2
Phoneme	71,82	5,8	76,41	11,5	79,66	17,4	75,55	3,6	82,14	17,8
Pima	73,97	7,4	73,71	7,8	75,26	10,6	73,37	3,1	75,66	22,7
Ring	50,62	13,8	79,63	4,6	86,92	6,9	72,63	6,8	94,08	24,0
Spambase	88,10	21,7	70,14	7,9	77,22	3,9	72,98	3,7	91,93	29,8
Two-norm	93,47	31,4	86,99	26,5	85,97	12	73,98	3,1	95,28	60,9
Wdbc	91,91	6,6	92,33	5,2	92,26	7,2	90,68	3,7	95,25	10,4
Cleveland	58,61	56,4	48,82	11,9	53,51	6,9	51,59	6,4	55,24	61,3
Glass	67,07	46,7	58,05	15,1	57,99	9,4	58,49	6,9	70,24	22,7
Iris	95,33	10,9	94,44	4,0	94,00	14,9	94,89	3,4	96,00	4,0
Page-blocks	94,68	11,8	91,39	7,5	94,21	7,4	90,72	6,5	95,01	19,1
Penbased	83,94	111,9	81,16	40,0	50,45	18,4	67,93	15,9	96,04	152,8
Satimage	81,55	61,8	81,69	57,9	74,72	16,5	77,10	12,2	87,32	76,1
Texture	87,24	74,8	81,57	34,9	70,15	14,6	71,66	18,6	92,89	54,5
Vowel	63,84	113,6	71,11	63,1	67,07	9,2	65,83	18	71,82	72,3
Wine	93,76	16,8	89,47	5,5	92,61	9,2	91,88	4,2	94,35	8,7

- [9] Oscar Cordón. 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.
- [10] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. 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.
- [11] M. Fazzolari, R. Alcalá, Y. Nojima, H. Ishibuchi, and F. Herrera. A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions. *Fuzzy Systems, IEEE Transactions on*, 21(1):45–65, 2013.
- [12] Alberto Fernández, María Calderón, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 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.
- [13] M. J. Gacto, R. Alcalá, and F. Herrera. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, 181(20):4340–4360, 2011.
- [14] F. Herrera. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1(1):27–46, 2008.
- [15] H. Ishibuchi and T. Yamamoto. Rule weight specification in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 13(4):428–435, 2005.
- [16] H. Ishibuchi, M. Yamane, and Y. Nojima. Rule weight update in parallel distributed fuzzy genetics-based machine learning with data rotation. In *IEEE International Conference on Fuzzy Systems, 2013. FUZZ-IEEE 2013*, pages 1–8. IEEE, 2013.
- [17] E. P. Klement, R. Mesiar, and E. Pap. *Triangular norms*. Kluwer Academic Publishers, Dordrecht, 2000.
- [18] A. S. Koshiyama, M. M. B. R. Vellasco, and R. Tanscheit. *GPFIS: Um Sistema Fuzzy-Genético Genérico baseado em Programação Genética*. Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2014.
- [19] Daniel Leite, Rosangela Ballini, Pyramo Costa, and Fernando Gomide. Evolving fuzzy granular modeling from nonstationary fuzzy data streams. *Evolving Systems*, 3(2):65–79, 2012.
- [20] Edwin Lughofer. On-line assurance of interpretability criteria in evolving fuzzy systems - Achievements, new concepts and open issues. *Information Sciences*, 251:22–46, 2013.
- [21] Thair Nu Phyu. Survey of Classification Techniques in Data Mining. *International Multiconference of Engineers and Computer Scientists*, 1:18–20, 2009.
- [22] L. X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, 22(6):1414–1427, 1992.