

# SÍNTESE DE ÁRVORES DE PADRÕES FUZZY ATRAVÉS DE BUSCA PARTICIPATIVA

ADRIANO V. DE BARROS<sup>1</sup>, JORGE L. M. DO AMARAL<sup>2</sup>, ANDERSON R DOS SANTOS<sup>3</sup>, DOUGLAS M. DIAS<sup>2</sup>

1. Programa de Pós-Graduação em Engenharia Eletrônica – PEL
  2. Departamento de Engenharia Eletrônica e de Telecomunicações – DETEL
  3. Laboratório de Redes Industriais e Sistemas de Automação – LARISA
- Universidade do Estado do Rio de Janeiro

R. São Francisco Xavier, 524 – Maracanã, Rio de Janeiro – RJ, 20550-900

E-mails: adriano.vbarrosavb@gmail.com, jamaral@uerj.br,  
andersonrds\_rj@hotmail.com, douglas.dias@uerj.br

**Abstract**— In this paper, it is proposed a new methodology for the elaboration of Fuzzy Pattern Trees using the junction of two algorithms. The first is a Participative Search method with Arithmetic Recombination that in the computational tests showed to be efficient, competitive and computationally simple, being, at first directed to numerical optimization problems and that uses the differential evolution as the basis of genetic operators. The second one is an approach that aims to use the interpretability of the Fuzzy models in the search for knowledge in a database maintaining the commitment to the accuracy, the Fuzzy Pattern Trees. This is a fuzzy model with a hierarchical tree structure that becomes interesting at several points and in a general way can be seen as a generalized logical description of a class. The model becomes a viable alternative to the classic model of Fuzzy rules. The join intends to use the efficiency presented in the Search Participation process to produce small and interpretable Fuzzy Pattern Trees with competitive accuracy levels. The results obtained with the proposed method showed that the performance achieved, using some data available in the UCI Machine Learning Repositor, brings greater interpretability while maintaining levels of accuracy within the state of the art.

**Keywords**— Participative Search, Fuzzy Systems, Fuzzy Pattern Trees, Optimization, Interpretability, Hierarchical Model, Classification.

**Resumo**— Neste trabalho é proposta uma nova metodologia para a elaboração de Árvores de Padrão Fuzzy utilizando a junção de dois algoritmos. O primeiro é um método de Busca Participativa com Recombinação Aritmética, que nos testes computacionais demonstrou ser eficiente, competitivo e computacionalmente simples, sendo a princípio direcionado para problemas de otimização numérica e que utiliza a Evolução Diferencial como a base dos operadores genéticos. O segundo é uma abordagem que visa utilizar a interpretabilidade dos modelos Fuzzy na busca por conhecimento em uma base de dados, mantendo o compromisso com a acurácia: as Árvores de Padrão Fuzzy. Este é um modelo Fuzzy hierárquico com uma estrutura em árvore que se torna interessante em diversos pontos e, de uma maneira geral, pode ser visto como uma descrição lógica generalizada de uma classe. O modelo se torna uma alternativa viável ao modelo clássico de regras Fuzzy. A junção pretende utilizar a eficiência apresentada no processo de Busca Participativa para elaborar Árvores de Padrão Fuzzy pequenas e interpretáveis com níveis de acurácia competitivos. Os resultados obtidos com o método proposto mostraram que o desempenho alcançado, utilizando algumas bases de dados disponíveis no UCI Machine Learning Repositor, traz maior interpretabilidade mantendo níveis de acurácia dentro do estado da arte.

**Palavras-chave**— Busca Participativa, Sistemas Fuzzy, Árvores de Padrão Fuzzy, Otimização, Interpretabilidade, Modelo Hierárquico, Classificação.

## 1 Introdução

O volume de dados disponíveis atualmente gera grande dificuldade de análise e estão além da capacidade de tempo disponível para fazê-lo. Um fator que se torna importante nessa análise é a extração de conhecimento desses dados, de forma que existe grande interesse no desenvolvimento de modelos que podem aprender a partir de uma base de dados. A extração de conhecimento pode se dar a partir da previsão do padrão comportamental dos dados. Porém, quando se deseja aliar a previsão com o entendimento da tomada de decisão, surgem algumas alternativas, como os modelos gerados por abordagens simbólicas. Os modelos de tal abordagem procuram unir sua interpretabilidade com níveis de acurácia competitivos. Alguns métodos mais bem-sucedidos aplicados na síntese de modelos interpretáveis são os baseados na teoria dos conjuntos *fuzzy*, pois esses modelos criam uma interface expressa em linguagem natural. A síntese de um sistema *fuzzy* baseado em regras não é simples e, no geral, de-

manda uma grande quantidade de regras quando se deseja um alto nível de acurácia, o que compromete sua interpretabilidade. Em contrapartida, um sistema com poucas regras possui altos níveis de interpretabilidade, porém sua acurácia pode ficar comprometida.

Os sistemas *fuzzy* baseados em regras podem representar tanto funções de classificação quanto de regressão. Existe um grande número de estratégias que foram desenvolvidas para induzir modelos *fuzzy* baseados em regras (Cordón, 2011).

Neste trabalho foi feito um sistema para indução de Árvores de Padrões Fuzzy (APF) de forma automática, utilizado na tarefa de classificação. Este método não é baseado em regras, mas é um método hierárquico, com uma estrutura de árvores não dependente de conhecimento anterior que propõe unir alta interpretabilidade com níveis elevados de acurácia. Este modelo foi apresentado pela primeira vez em (Huang et al., 2008). Na síntese das APFs, o método de aprendizado foi substituído por uma nova classe de algoritmo de busca baseado na aprendizagem participativa. O algoritmo de busca que se pretende utilizar é o

Busca Participativa com Recombinação Aritmética (*Participatory Search with Arithmetical Recombination - PSAR*). Ele demonstra ter um desempenho superior a diversos métodos do estado da arte, além de apresentar simplicidade, rapidez e um baixo número de parâmetros, tornando-o muito atraente.

Este trabalho está dividido da seguinte forma: na Seção 2 será apresentado o conceito das Árvores de Padrões Fuzzy; na Seção 3 será apresentada a Busca Participativa com Recombinação Genética e na Seção 4 será apresentado o método proposto com as decisões tomadas durante o trabalho desenvolvido. Por fim, a Seção 5 apresentará os resultados obtidos.

## 2 Árvores de Padrão Fuzzy - APF

As Árvores de Padrões Fuzzy são modelos hierárquicos com uma estrutura de árvore, na qual os nós internos são operadores utilizados nos sistemas *fuzzy* e as folhas dessas árvores são compostas por termos *fuzzy* associados a um atributo de entrada. A APF propaga a informação do fundo para o topo; os nós internos recebem os valores de seus antecessores e os combinam utilizando um operador, apresentando a saída para o nível superior. As APFs implementam um mapeamento recursivo, apresentando a saída em um intervalo unitário. Um classificador baseado em APF é construído criando uma árvore para cada classe. São inseridos os valores dos atributos que se deseja classificar nas entradas das árvores de cada classe e a predição da classe à qual esse conjunto de dados pertence é feita escolhendo-se a árvore que tem o maior valor de saída. Geralmente, cada árvore pode ser considerada como uma “descrição lógica” da classe, permitindo uma interpretação mais concreta do problema de aprendizado (Senge et al., 2011).

As APFs se apoiam em três conceitos: a fuzzificação dos atributos de entrada; a estruturação hierárquica de uma dependência funcional através do particionamento recursivo de um critério em subcritérios e a agregação flexível de subcritérios através de operadores *fuzzy* parametrizados (Senge et al., 2011).

Como apresentado na Figura 1, a associação entre um atributo e um termo *fuzzy* é representada por uma função de pertinência. Na Figura 1 é dado um exemplo de valor de entrada  $a_1$  dentro do universo de discurso deste atributo e seu valor fuzzificado  $\mu_1$ .

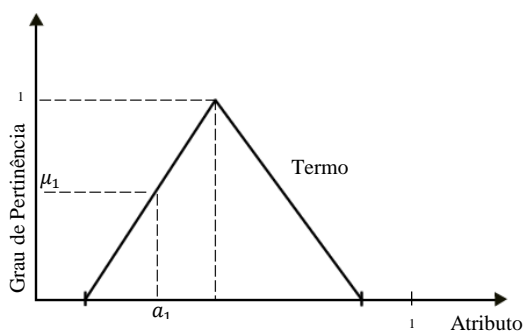


Figura 1: Associação de Atributo - Termo Fuzzy.

O modelo baseado em árvore de padrões mapeia diversas entradas em apenas uma variável de saída. A interpretação da saída produzida pode ser vista como um modelo que simplifica a avaliação global de uma propriedade, avaliando diferentes subcritérios e agregando essas avaliações posteriormente (Senge et al., 2011).

Um exemplo de APF pode ser visto na figura 2, onde a árvore apresentada determina uma classe que representa a qualidade de um vinho.

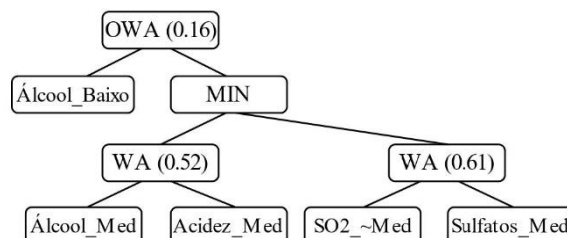


Figura 2: Árvore de Padrão Fuzzy para classificar a qualidade de vinhos.

Analisando a Figura 2, pode-se observar que a qualidade do vinho está relacionada com duas sub-árvores. A primeira relaciona o teor alcoólico e a acidez e a segunda sub-árvore trata da concentração de sulfato e de dióxido de enxofre, sendo que essas duas primeiras árvores são combinadas em um nível mais alto. No nível acima, o resultado da avaliação das duas sub-árvores é combinado com o nível alcoólico. A combinação realizada em todos os níveis da árvore é realizada utilizando t-normas, t-conormas e dois operadores de média. O operador de média (OWA - *Ordered Weighted Average*) é a combinação de  $k$  números,  $v_1, v_2, \dots, v_k$ , definido por

$$OWA_W(v_1, v_2, \dots, v_k) \stackrel{\text{def}}{=} \sum_{i=1}^k w_i v_{\tau(i)}, \quad (1)$$

onde  $\tau$  é uma permutação entre o conjunto de números  $\{1, 2, \dots, k\}$  de forma que  $V_{\tau(1)} \leq V_{\tau(2)} \leq \dots \leq V_{\tau(k)}$  e  $w = (w_1, w_2, \dots, w_k)$  é um vetor de pesos que satisfaz  $w_i \geq 0$  para  $i = 1, 2, \dots, k$ , onde o somatório dos pesos de 1 a  $k$  deverá ser igual a 1. O outro operador de média é o WA (*Weighted Average*), que é similar ao OWA, porém sem o ranqueamento dos valores de  $v$ .

O primeiro método de aprendizado é chamado de *Bottom-up Induction* (Huang et al., 2008). Nele a indução das árvores procura criar uma representação lógica para cada classe de uma forma iterativa. O processo ocorre do fundo para o topo. Em cada iteração do processo, combinam-se as duas melhores árvores candidatas com um novo nó interno. O uso dessa estratégia é motivado intuitivamente. Ela pode ser vista como uma combinação iterativa para a construção de características complexas, a partir de características mais básicas criadas pelos atributos originais (Senge et al., 2011).

Outro método para o aprendizado de FPTs é chamado *Top-down Induction* (Senge et al., 2011). Nessa estratégia, ao invés de juntar duas árvores em uma

nova árvore, muito maior e com certa diferença na estrutura, a ideia é fazer pequenas modificações. Isso é feito expandindo um nó de folha. Este procedimento de realizar pequenas modificações na árvore a cada iteração proporciona uma maior exploração do espaço de busca, o que aumenta a chance de encontrar uma árvore que atenda ao requisito do usuário.

A estratégia *top-down* possui algumas desvantagens: A primeira delas está relacionada ao fato do algoritmo de busca (*beam search*) utilizado ter uma característica “gulosa”, olhando sempre para o melhor candidato no estágio atual de construção. Essa característica às vezes pode fazer com que o espaço de busca seja restringido, fazendo com que o algoritmo fique preso em sub-ótimos globais. A outra desvantagem está ligada à maldição da dimensionalidade. Se a quantidade de atributos for grande, e a largura do feixe (*beam*) também for, o algoritmo terá problemas em testar todas as possibilidades, pois haverá uma explosão na quantidade de possibilidades. Por outro lado, se a largura do feixe for pequena, então apenas uma pequena região do espaço de busca será explorada, o que pode fazer com que o algoritmo obtenha soluções insatisfatórias (Santos, 2014).

### 3 Busca Participativa com Recombinação Aritmética

O processo de busca é útil em inúmeras aplicações como otimização de funções, aprendizagem de máquina, processamento e recuperação da informação.

No caso dos algoritmos baseados no paradigma de aprendizado de busca participativa, a busca é guiada pela compatibilidade entre indivíduos de uma população, sempre mantendo o melhor indivíduo nas populações posteriores e introduzindo indivíduos de forma aleatória em cada etapa do algoritmo. A operação de recombinação é realizada através de uma combinação convexa regulada pela compatibilidade entre os indivíduos. Já a operação de mutação é similar à aquela encontrada na Evolução Diferencial (*Differential Evolution* - DE) (Price et al., 1997), e também leva em conta a compatibilidade entre indivíduos selecionados e recombinações.

Um fator fundamental da aprendizagem participativa é o grau de compatibilidade entre a informação de entrada e o conhecimento atual. Uma questão é que nenhuma facilidade é fornecida para medir a confiança que temos na estrutura de conhecimento atual. Isso significa que à medida que avançamos é como se tivéssemos confiança completa em nosso conhecimento atual. Se uma longa sequência de entradas tem pouca compatibilidade com o conhecimento atual, podemos acreditar que o que foi aprendido até agora é errado. Isso é visto como uma forma de excitação. O aprendizado participativo introduz um mecanismo de excitação para monitorar o desempenho do processo de aprendizagem, observando os valores do grau de compatibilidade do conhecimento atual com os insu-  
mos (Ling, 2016).

Em (Ling, 2016), é apresentada uma coleção de procedimentos de aprendizagem construídos com base na computação evolutiva e na aprendizagem participativa, centrando a pesquisa na Busca Participativa com Recombinação Aritmética (*Participatory Search with Arithmetical Recombination* - PSAR). O trabalho introduziu uma nova classe de algoritmos de pesquisa baseados em população usando operadores participativos de seleção, recombinação e mutação. Em conjunto, os graus de compatibilidade e a função objetivo moldam a avaliação da dependência dos indivíduos e governam a evolução da população. A recombinação de PSAR emerge de uma instância de uma fórmula de atualização de aprendizagem participativa e se assemelha ao cruzamento aritmético, exceto que é modulado por um grau de compatibilidade e um índice de excitação. A mutação no PSAR também depende do grau de compatibilidade e do índice de excitação da aprendizagem participativa. Isso decorre de um esquema semelhante à mutação da DE.

Os resultados computacionais corroboram que o algoritmo é competitivo, computacionalmente simples e eficiente. As aplicações da pesquisa participativa na modelagem baseada em regras difusas usando dados reais, em (Ling, 2016), ilustram o potencial da pesquisa participativa para resolver problemas complexos. Os resultados computacionais em (Ling, 2016), também sugerem que o algoritmo de pesquisa participativa funciona melhor do que a abordagem de sistema *fuzzy* genético de estado atual da arte. Além do seu desempenho, do ponto de vista da qualidade das soluções que eles desenvolvem, os algoritmos de pesquisa participativa são mais simples, são mais rápidos e requerem menos parâmetros, o que os torna muito atraentes na prática.

A seguir, as Seções 3.1 a 3.4, serão brevemente apresentados e revisados os principais pontos do algoritmo PSAR; uma descrição mais detalhada pode ser vista em (Ling, 2016).

#### 3.1 Compatibilidade

Como um fator fundamental no conceito da aprendizagem participativa que expressa o conhecimento atual e a entrada atual para atualizar o conhecimento, é usado um grau de compatibilidade da seguinte forma:

$$v(t+1) = v(t) + \alpha \rho_t (z(t) - v(t)), \quad (2)$$

onde  $z(t)$  e  $v(t)$  são vetores  $n$ -dimensionais que indicam o conhecimento atual e a entrada atual, respectivamente. Assumimos, sem perda de generalidade, que  $v(t)$  e  $z(t) \in [0,1]^n$ . O parâmetro  $\alpha \in [0,1]$  é a taxa de aprendizagem básica e  $\rho_t \in [0,1]$  é o grau de compatibilidade entre  $z(t)$  e  $v(t)$  no passo  $t$ . O produto da taxa de aprendizado básico pelo grau de compatibilidade produz a taxa de aprendizado efetiva. Se uma entrada está longe do conhecimento atual, o valor do grau de compatibilidade correspondente é pequeno e a entrada é filtrada, uma vez que a taxa de aprendizado é reduzida pelo grau de compatibilidade. Isso significa

que se os dados de entrada estiverem em grande discordância e confusos com o conhecimento atual, eles serão descontados. A maneira adotada para calcular o grau de compatibilidade  $\rho_t$  no passo  $t$  é dada por:

$$\rho_t = 1 - \frac{1}{n} \sum_{k=1}^n |z(t) - v(t)| \quad (3)$$

Em um sentido mais geral,  $\rho_t$  pode ser visto como uma medida de semelhança entre  $z(t)$  e  $v(t)$ . Se  $\rho_t = 0$ , então  $v(t+1) = v(t)$  e a entrada atual  $z(t)$  são completamente incompatíveis com o conhecimento atual  $v(t)$ . Essa condição significa que o sistema não está aberto a qualquer aprendizado com as informações atuais. Por outro lado, se  $\rho_t = 1$ , então  $v(t+1) = z(t)$ . Nesse caso, a informação de entrada está totalmente de acordo com os conhecimentos atuais e o sistema está totalmente aberto para aprender.

### 3.2 Seleção

Seja  $S$  um conjunto de  $N$  vetores de comprimento fixo  $n$ , e  $s, s' \in S$  sejam dois indivíduos,  $s'$  distinto de  $s$ , de modo que:

$$s' = \operatorname{argmax}_{r \in S} (\rho(s, r)), \quad (4)$$

onde

$$\rho(s, r) = 1 - \frac{1}{n} \sum_{k=1}^n |s_k - r_k| \quad (5)$$

e  $s = (s_1, s_2, \dots, s_N)$  e  $r = (r_1, r_2, \dots, r_N)$ . O indivíduo  $s'$  é aquele cujo índice de compatibilidade com  $s$  é o maior. Este procedimento é repetido para cada indivíduo  $s$  de  $S$  para montar a população  $S'$  com  $N$  indivíduos. Observe que a obtenção dessa população é tendenciosa pelos graus de compatibilidade entre os indivíduos de  $S$ .

A seleção é feita calculando os graus de compatibilidade entre  $s \in S$  e o  $s' \in S'$  correspondente com o melhor  $best = s^*$  atual. A população  $L$  de indivíduos selecionados é obtida escolhendo o indivíduo de uma das duas populações que seja mais compatível com o melhor indivíduo atual. A Figura 3 ilustra o processo de seleção e as seguintes equações definem formalmente o melhor  $s^*$  e a compatibilidade:

$$s^* = \operatorname{argmin}_{s \in S} (f(s, r)) \quad (6)$$

$$\rho^{s'} = 1 - \frac{1}{n} \sum_{k=1}^n |s'_k - s_k^*| \quad (7)$$

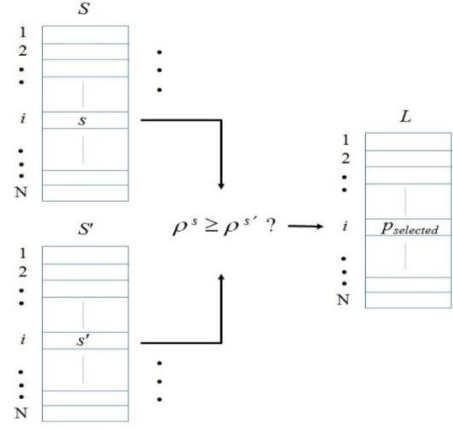


Figura 3: Seleção (Ling, 2016).

A seleção depende da função de aptidão  $f(s)$ , que identifica o melhor  $s^*$  atual, e em  $\rho^s(s, s^*)$  e  $\rho^{s'}(s', s^*)$  que medem a compatibilidade entre  $s^*$  e o par correspondente de indivíduos e  $s'$  da população atual. Em conjunto,  $f, \rho^s$  e  $\rho^{s'}$  decidem se um indivíduo está selecionado ou não (Ling, 2016).

### 3.3 Recombinação

A recombinação deriva da fórmula de atualização de aprendizagem participativa de acordo com a equação:

$$v(t+1) = v(t) + \alpha \rho_t^{1-a_t} (z(t) - v(t)). \quad (8)$$

Observe que a fórmula da atualização de aprendizagem participativa pode ser reescrita como:

$$v(t+1) = (1 - \alpha \rho_t^{1-a_t})v(t) + (\alpha \rho_t^{1-a_t})z(t) \quad (9)$$

A recombinação participativa prossegue como na equação 9 para produzir o  $p_r$  de indivíduos  $s$  e  $s'$  dos grupos  $S$  e  $S'$ , respectivamente, da seguinte forma:

$$p_r = (1 - \alpha \rho_t^{1-a_t})s + (\alpha \rho_t^{1-a_t})s' \quad (10)$$

onde  $a_t \in [0,1]$  é o índice de excitação. A maneira de modelar a excitação é vê-la como o complemento da confiança no conhecimento atual em um procedimento simples, atualizando o índice de excitação a cada etapa  $t$  da seguinte forma:

$$a_{t+1} = (1 - \beta)a_t + \beta(1 - \rho_{t+1}) \quad (11)$$

onde o parâmetro  $\beta \in [0,1]$  controla a taxa de mudança de excitação.

### 3.4 Mutação

Existem muitas maneiras de fazer mutação nos algoritmos de pesquisa. A mutação na pesquisa participativa é semelhante à mutação de evolução diferencial. Produz uma  $p_m$  individual mutada de acordo com a equação:

$$p_m = best + \rho_t^{1-a_t} (p_{selected} - p_r). \quad (12)$$

#### 4 Método Proposto

Este trabalho propõe a utilização do algoritmo de busca participativa como algoritmo de síntese de APFs. Dentre as razões para esta escolha pode-se citar:

- A exploração do espaço de busca na busca participativa não segue a estratégia gulosa dos métodos convencionais e, portanto, tem mais chances de obter soluções melhores.
- A exploração do espaço não é feita de forma restrita (que é limitada pela largura do feixe). Isso também aumenta as chances de obter soluções melhores.
- O desempenho do PSAR obtido a partir de experimentos computacionais extensivos demonstra ser superior às diversas instâncias de busca no estado da arte atual, o que o torna interessante no processo de síntese das APFs.
- O PSAR apresenta um baixo número de parâmetros (tamanho da população e número de gerações).

O PSAR é um método de otimização numérico que não se encaixa diretamente na representação das APFs devido aos seus operadores de recombinação e mutação se darem de forma aritmética. Dessa forma este trabalho propõe alterações nesses operadores em específico. Outra alteração proposta é a modificação na compatibilidade entre indivíduos, que segue método semelhante à Distância de Edição (Navarro, 2001).

O modelo proposto extrai a compatibilidade e o método de seleção utilizado no PSAR e altera os operadores de recombinação e mutação, substituindo-os, inicialmente, por operadores simples de cruzamento e mutação do Algoritmo Genético Clássico (Holland, 1975; 1992).

Para o modelo proposto é necessária uma medida de compatibilidade entre dois indivíduos de mesmo tamanho, de forma que  $\rho \in [0,1]$ . Sendo assim, propõe-se uma medida que avalie a quantidade de posições iguais dentro do cromossomo de números inteiros em proporção ao seu tamanho. Na Figura 4 é ilustrado o cálculo de compatibilidade entre vetores  $I_1, I_2$  (cromossomos), de dimensão 9 e de acordo com o método proposto.

$$\begin{aligned}
 I_1 &= [1 \ 3 \ 2 \ 6 \ 9 \ 8 \ 7 \ 4 \ 5] \\
 I_2 &= [1 \ 2 \ 3 \ 6 \ 9 \ 4 \ 7 \ 8 \ 5] \\
 &\quad \downarrow \quad \downarrow \downarrow \downarrow \downarrow \\
 S &= [1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1] \\
 \rho &= \frac{(\sum_{k=1}^9 S(k))}{9} = \frac{5}{9}
 \end{aligned}$$

Figura 4: Exemplo de cálculo de compatibilidade.

A função de aptidão proposta para o modelo é composta por duas parcelas, acurácia e interpretabilidade. Ela representa uma forma de função multiobjetivo, sendo composta pela ponderação entre as parcelas envolvidas, assim como apresentado na equação:

$$Aptidão = f(s, r) = w_1 \times AP_1 + w_2 \times AP_2 \quad (13)$$

Na equação 12,  $w_1$  e  $w_2$  são pesos definidos antes da inicialização do algoritmo e determinam a proporção entre acurácia e interpretabilidade, desde que  $w_1 + w_2 = 1$ .

A parcela  $AP_1$  é estabelecida pela equação 14, onde o RMSE (*Root Mean Square Error* – raiz quadrada do erro quadrático médio) é definido a partir da diferença entre o valor obtido na saída da árvore e o valor alvo, que deve ser 1 se os valores apresentados na entrada pertencem a um ponto da classe que a árvore deve representar e 0, caso contrário.

$$AP_1 = 1 - RMSE \quad (14)$$

A parcela  $AP_2$  é calculada pela equação 15, sendo inversamente proporcional à quantidade de posições ativas dentro do cromossomo ( $Q_{genesAtivos}$ ) em comparação à quantidade total de posições ( $Q_{genes}$ ). Dessa forma, árvores menores tem uma melhor avaliação em comparação às mais longas.

$$AP_2 = \frac{(Q_{genes} - Q_{genesAtivos})}{Q_{genes}} \quad (15)$$

No modelo desenvolvido a representação das árvores se baseia em PG Cartesiana (Miller and Thonson, 2000), sendo denominados cromossomos; alguns exemplos podem ser vistos na Figura 5. Esses cromossomos podem ser divididos em partes, denominadas genes, que, por sua vez, podem ser divididos em genes de função, conexão ou saída.

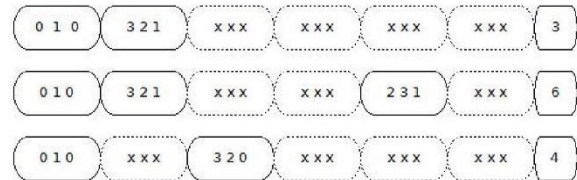


Figura 5 – Cromossomos (adaptado de Santos, 2014).

A junção de um gene de função com alguns genes de conexão forma um nó. A estrutura de cada nó consiste em uma função (gene de função) em que são aplicados os valores de entrada, cuja localização é dada pelo gene de conexão. Para gerar uma saída, a função é escolhida dentro de um conjunto de funções previamente definidas de acordo com a aplicação. A entrada de cada nó pode ser uma entrada do sistema ou a saída de um nó de uma coluna anterior (Miller, 2011).

Quando o cromossomo é decodificado, alguns nós podem estar desativados; isso acontece porque al-



guns deles não estão ligados à saída de dados. Na Figura 5 os nós desativados são representados pela ausência de valores com a marcação 'x'.

## 5 Resultados

Serão apresentados alguns experimentos iniciais realizados para avaliar o desempenho do modelo criado, através da comparação com outros classificadores bastante conhecidos da área de aprendizado de máquina.

Foram utilizados para testar e comparar o algoritmo as bases de dados IRIS, WINE, SONAR e PIMA, que estão disponíveis no UCI (machine learning repository). Também foram utilizadas duas bases de dados artificiais DIFFICULT e BANANA geradas a partir da toolbox PRTools (van der Heijden et al. 2004) em ambiente Matlab. A base de dados Iris possui 150 objetos, 3 classes e 4 atributos. A base de dados Wine possui 178 objetos, 3 classes e 13 atributos. A base de dados Sonar possui 208 objetos, 2 classes e 60 atributos. A base de dados Pima possui 768 objetos, 2 classes e 8 atributos. Ambas as bases de dados artificiais foram utilizadas com 1.000 pontos, 2 classes e 2 atributos.

Os algoritmos utilizados para comparações foram: K-Vizinhos mais próximos (KNN) (Webb, 2002), florestas aleatórias (RF: Random forests) (Breiman, 2001), APF1 e APF2 (Santos, 2014). Os algoritmos KNN e RF foram escolhidos pois são conhecidos por terem um bom desempenho em um grande número de *benchmarks* e foram implementados com a toolbox PRTools (van der Heijden et al. 2004). Os algoritmos APF1 e APF2 também serão utilizados pois apresentam resultados competitivos na síntese de APF. A estimativa do desempenho de generalização foi feita através de uma validação cruzada com cinco partições e cinco repetições.

Foram utilizadas duas medidas de desempenho: a taxa de acertos na classificação e a área sob a curva ROC.

Os parâmetros dos métodos APF1 e APF2 são os mesmos descritos em (Santos, 2014), com 100 e 4.000 nós, respectivamente. O critério de parada foi definido em um total de 600 gerações com  $w_1 = 0,7$  e  $w_2 = 0,3$  em ambos casos. Os parâmetros dos outros classificadores foram determinados pelas configurações default do PRTools: O número de vizinhos K foi fixado em 1; o número de árvores geradas nas *Random Forests* foi igual a 50 e o número de atributos sob o qual é feita a partição foi igual a 1. Os parâmetros do método proposto foram estabelecidos da seguinte forma: o critério de parada foi definido pelo número de gerações igual a 150; já os demais parâmetros ( $w_1$ ,  $w_2$ , número de nós e tamanho da população) foram determinados através de uma busca em *grid* observando curvas de aprendizado com 5 repetições de experimentos para cada proporção dos grupos. Um exemplo de curva de aprendizado observada pode ser visto na Fi-

gura 5, onde busca-se evitar erros de *bias* e de variância analisando a convergência do erro do grupo de treinamento e teste.

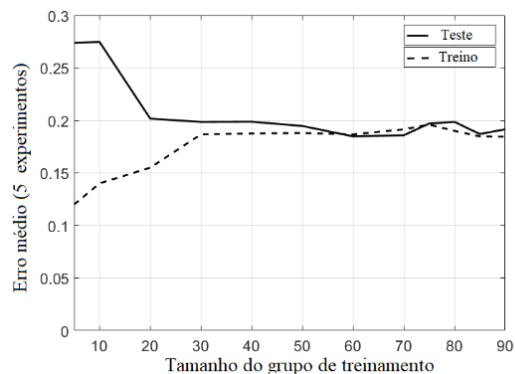


Figura 5 – Curva de aprendizado

### 5.1 Análise dos resultados

Os resultados podem ser vistos na Tabela 1, que mostra o erro de acurácia e o desvio de padrão, respectivamente.

É possível notar que o método proposto (BP\_APF) tem resultados próximos aos dos demais utilizados na comparação. Observa-se que o APF2 com um maior número de nós possui um desempenho melhor em relação ao APF1, o que pode ser atribuído a um melhor aproveitamento da característica da neutralidade. Porém, esta melhoria no desempenho da classificação vem em troca de um aumento da árvore gerada, o que significa uma maior dificuldade de explicação e compreensão da expressão obtida. Os resultados de APF1 e APF2 corroboram aqueles obtidos nos trabalhos já desenvolvidos na síntese das APFs (Santos, 2014).

Em todos os resultados obtidos utilizando o BP\_APF, as árvores obtidas apresentaram profundidade reduzida em relação aos métodos APF1 e APF2, o que gerou maior interpretabilidade uma vez que podemos definir qual termo *fuzzy* influencia mais significativamente no resultado. Não foram alcançadas árvores com profundidade superior a três, sendo em sua maioria obtidas árvores de profundidade dois e um.

Tabela 1: Erro de classificação

Base de Dados	BP_APF	KNN	RF	APF1	APF2
Iris	0,06±0,01	0,04±0,003	0,05±0,008	<b>0,04±0,01</b>	0,04±0,07
Wine	0,07±0,01	0,27±0,03	<b>0,01±0,003</b>	0,08±0,01	0,06±0,07
Sonar	0,26±0,01	0,16±0,01	<b>0,13±0,01</b>	0,24±0,02	0,24±0,01
Pima	<b>0,25±0,006</b>	0,32±0,001	0,25±0,008	0,26±0,01	0,25±0,01
Banana	0,19±0,004	0,18±0,01	<b>0,16±0,01</b>	0,22±0,01	0,17±0,01
Difficult	0,28±0,01	0,22±0,01	<b>0,21±0,01</b>	0,33±0,03	0,31±0,03

Foi utilizado o teste de Friedman (Friedman, 1937, 1940) a partir da Tabela 1 para determinar se há diferença estatística significativa entre os métodos experimentados, sendo utilizado neste teste o parâmetro

$\alpha = 0,05$ . O resultado apresentado determina que a hipótese nula  $H_0$  deve ser rejeitada indicando diferença entre os métodos. Junto ao teste Friedman foi utilizando o método Simes-Hochberg com nível de significância de 0,05 para gerar uma comparação múltipla entre métodos. Foi constatado que somente existe diferença significativa entre os métodos BP\_APF e KNN.

## 6 Conclusão

Neste Artigo foi proposto um modo alternativo para sintetizar as Árvores de Padrões Fuzzy, baseado na Busca Participativa com Recombinação Aritmética como algoritmo de aprendizado. Os resultados obtidos mostram que o desempenho do método na tarefa de classificação pode se aproximar do desempenho de alguns classificadores consolidados disponíveis na literatura. O método também introduz um modelo mais interpretável em relação aos outros métodos de síntese das Árvores de Padrão Fuzzy, o que o torna transparente ao conhecimento obtido no aprendizado. Dessa forma, fornece com mais facilidade informações úteis na distinção dos atributos mais significativos da base de dados em análise, mantendo níveis competitivos de desempenho. As Árvores de Padrões Fuzzy tornam-se uma alternativa viável aos modelos clássicos *fuzzy* baseados em regras, pois sua estrutura hierárquica permite uma representação mais compacta e um compromisso entre acurácia e a simplicidade do modelo. Algumas sugestões para trabalhos futuros, a partir da abordagem proposta, se estabelecem em torno dos seguintes itens: aperfeiçoamento da função multiobjetivo utilizando o conceito de dominância de Pareto (Zhou et al., 2011); aplicação de um subsistema *fuzzy* como operador. Neste sistema baseado em regras, haveriam duas entradas e uma saída conforme (Zhang, 2006) ou na forma de Quadtree; explorar o desempenho do método proposto utilizando outros operadores de recombinação e mutação; aperfeiçoar a codificação dos indivíduos da população de forma que o PSAR (Ling, 2016) possa ser utilizado com menos modificações na busca pelas melhores APF; extensão do modelo para tarefas de controle.

## Agradecimentos

Agradecemos à Capes e Fapetj pelo apoio prestado neste projeto.

## Referências Bibliográficas

- Breiman, L., 2001. Random Forests. *Mach. Learn.* 45, 5–32. doi:10.1023/A:1010933404324
- Cordón, O., 2011. A Historical Review of Evolutionary Learning Methods for Mamdani-type Fuzzy Rule-based Systems: Designing Interpretable Genetic Fuzzy Systems. *Int J Approx Reason.* 52, 894–913. doi:10.1016/j.ijar.2011.03.004
- Friedman M., 1937 The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*; 32: 675–701.
- Friedman M., 1940. A comparison of alternative tests of significance for the problem of  $m$  rankings. *Annals of Mathematical Statistics*; 11; 86–92.
- van der Heijden, F., R. P. W. Duin, D de Ridder and D. M. J. Tax. 2004. Classification, parameter estimation and state estimation: Na engineering approach using MATLAB. New York: Wiley.
- Holland, J.H. 1975 “Adaptation in Natural and Artificial Systems”, University of Michigan Press.
- Holland, J.H. 1992 “Adaptation in Natural and Artificial Systems”, 2nd edition, The MIT Press.
- Hüllermeier, E. 2005. Fuzzy Methods in Machine Learning and Data Mining: Status and Prospects. *Fuzzy Sets Syst* 156, 387–406. doi:10.1016/j.fss.2005.05.036
- Ling, L.Y. 2016. Participatory Search Algorithms and Applications. Doctorate thesis, School of Electrical and Computer Engineering, University of Campinas – SP, Brazil.
- Miller, J. F.; Thonson, P., 2000. Cartesian Genetic Programming. In: POLI, R. et al. (Eds.). Genetic Programming. Lecture Notes in Computer Science. [s.l.] Springer Berlin Heidelberg. p. 121–132.
- Navarro, G., A Guided Tour to Approximate String Matching, *ACM Computing Surveys*, vol. 33, no. 1, pp. 32-88, Mar. 2001
- Price, K. R., Storn, R. M., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, Decem
- Santos, A.R., Amaral, J.L.M 2014. Síntese de Árvores de Padrão Fuzzy Através de Programação Genética. XX Congresso Brasileiro de Automática. Belo Horizonte – MG.
- Senge, R., Hüllermeier E., 2011. Top-Down Induction of Fuzzy Pattern Trees. *Trans Fuz Sys* 19, 241–252. doi:10.1109/TFUZZ.2010.2093532
- Webb, A.R., 2002. Density Estimation – Nonparametric, in: *Statistical Pattern Recognition*. John Wiley & Sons, Ltd, pp. 81–122.
- Yager, R., 1990. A model of participatory learning. *Man and Cybernetics*, IEEE Transactions on Systems, 20(5):1229-1234.
- Zhang, X., & Zhang, N., 2006. Universal approximation of binary-tree hierarchical fuzzy systems with typical FLUs. In *International Conference on Intelligent Computing* (pp. 177–182). Springer, Berlin, Heidelberg.
- Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q., 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), 32-49.