# Aplicando método do gradiente ótimo na otimização do cálculo do grau de cobertura das regras em árvores de decisão Fuzzy

Luiz Antônio Ferreira do Nascimento Junior<sup>1</sup>

**Resumo**: Esse trabalho tem como objetivo realizar um levantamento na literatura sobre as técnicas aplicadas para confecção de arvores de decisão fuzzy e comparar algumas delas com o intuito de analisar sua performance para comprovar a aplicabilidade do método de otimização no cálculo de cobertura de cada regra gerada através de arvores de decisão fuzzy nos mais diversos problemas existentes no mundo real. Os algoritmos abordados neste trabalho são o FID3, FuzzyDT, FCART, FlexDT e uma versão modificada do FuzzyDT nomeada FuzzyDTC.

Palavras-chave: cobertura, fuzzy decision tree, gradiente, id3, pertinência

**Abstract:** This study aims to survey the literature on the techniques used for making fuzzy decision trees and compare some of them in order to analyze its performance to prove an applicability of optimization method without each rule coverage calculation generated through Fuzzy Decision Trees in the most diverse problems in the real world. The algorithms covered in this work are FID3, FuzzyDT, FCART, FlexDT and a modified version of FuzzyDT named FuzzyDTC.

Keywords: cover, fuzzy decision tree, gradient, id3, pertinence

# 1 Introdução

Arvore de decisão é um método bem difundido na literatura que busca predizer novos casos baseados em casos antigos. A ideia básica envolve uma abordagem em que se trabalha com várias etapas para se dividir um problema complexo em vários problemas menores com o intuito de se obter através da união destes, a solução para o problema original. Algoritmos que exploram esse conceito, como o ID3 [1] e o C4.5 [2] por exemplo tem sido cada vez mais explanados e novas técnicas de otimização tem sido encontradas para solução de problemas do cotidiano.

O principal objetivo na utilização de aprendizagem de máquina na verdade é derivar padrões de uma quantidade limitada de dados. Hoje, com o continuo avanço da tecnologia digital, cada vez mais informação é produzida e armazenada. Entretanto, nem todos estes dados são úteis para o aprendizado de máquina, porque neles geralmente estão incluídos juntos aos dados uteis, três outros tipos de dados [3], como os ruidosos, redundantes e incompletos.

Arvores de decisão podem não ter um desempenho satisfatório em uma base de dados grande onde esses três tipos de dados sejam potencializados, dessa maneira a adaptação da lógica fuzzy a esse modelo visa reduzir os impactos da qualidade dos dados [3]. Algoritmos de arvores de decisão fuzzy têm sido propostos a fim de proporcionar uma forma de lidar com incerteza nos dados coletados, já que na literatura encontram-se vários exemplos demonstrando sua superioridade em relação a algoritmos de arvores de decisão crisp como em [4], [5] e [6].

{luizjr@eng-ele.mest.ufmg.br,lafnjr@gmail.com}

http://dx.doi.org/10.5335/rbca.v9i3.6324

<sup>&</sup>lt;sup>1</sup> Laboratório, PPGEE, Universidade Federal de Minas Gerais - Avenida Antônio Carlos, 6627 - Escola de Engenharia - Bloco I sala 2303

Este trabalho tem como objetivo realizar um estudo comparativo entre algumas técnicas de arvore de decisão Fuzzy (nebulosa) comparando os resultados obtidos de cada algoritmo na aplicação a diversas bases de dados diferentes a fim de comparar a eficiência do método do gradiente ótimo na otimização do cálculo do grau de cobertura visando uma otimização da predição do modelo, adquirindo uma maior generalização.

Uma breve revisão de algumas das técnicas encontradas na literatura é explanada na seção 2 desse artigo; Na seção 3 abordamos uma explicação sobre como o experimento foi planejado; Na seção 4 temos os resultados e na seção 5 apresentamos as conclusões finais.

## 2 Trabalhos correlatos

Existem vários trabalhos abordando modelos diversos de arvores de decisão fuzzy. Em [7], é apresentado uma introdução do conceito de arvores de decisão fuzzy, onde também aborda a otimização do método utilizando como base algoritmos genéticos. Uma abordagem empregando particionamento de atributos fuzzy utilizando uma versão chamada FID3 é utilizada em [8]. Em 2008, [9] propõe um classificador hibrido de arvores de decisão fuzzy com algoritmos evolucionários. [10] e [11] criou uma versão de um algoritmo de arvore de decisão fuzzy chamada FuzzyDT, que mostrou-se uma versão com baixa taxa de erros, sendo escolhida como ponto de partida para este trabalho devido sua simplicidade. Foram ainda considerados outros métodos que lidam com algoritmos de arvore de decisão fuzzy como em [3], [12] e [13].

Este trabalho contribui realizando uma análise comparativa entre os algoritmos citados e ainda propõe uma metodologia diferente no cálculo do Grau de Cobertura da Regra, na hora da realização da poda, como é próprio dos algoritmos baseados no C4.5 com a finalidade de aumentar a generalização do classificador fuzzy.

# 3 Revisão bibliográfica

## 3.1 Entropia

A entropia de Shannon [14] é utilizada na mensuração de impurezas dos dados crisp e é proposta da seguinte forma em (1):

$$Entropia(E_0) = -\sum_{i=1}^{L} p_i log_2 p_i$$

Sendo que para conjuntos crisp  $E_0 \subseteq E$ , a entropia é definida conforme (1), enquanto  $p_i$  é a proporção do número de elementos ao longo da classe  $C_i$  em relação ao número total de todos elementos em  $E_0$ .

Claramente, a entropia aumenta à medida que a proporção de todas as classes consegue se equiparar. Quando todos os elementos em E0 pertencem à mesma classe, a entropia é mínima; quando elementos de todas as classes têm a mesma proporção, entropia atinge o seu máximo [3].

#### 3.2 Grau de cobertura

O grau de cobertura de uma regra é definido a partir da somatória dos graus de compatibilidade dos exemplos do conjunto de treinamento quando classificados somente pela regra em questão. O grau de cobertura é calculado apenas uma vez, antes do início da execução dos algoritmos. Assim, regras que não classificam nenhum exemplo do conjunto de treinamento podem ser descartadas na definição do espaço de busca de soluções.

Seja R =  $\{R1, R2, ..., Rm\}$  o conjunto de m regras possíveis para o problema. Seja a regra Rt =  $\{ap1, ap2, ..., apn\}$ , formada por n antecedentes, uma das regras de R. Seja E =  $\{e1, e2, ..., eq\}$  o conjunto com q

exemplos do problema a ser tratado contendo valores numéricos associados ás variáveis linguísticas definidas para o problema. O grau de cobertura (GdC) de cada regra é calculado seguindo os seguintes passos:

- 1. Calcular o grau de compatibilidade GdC(Ri,ei) do exemplo e1 com a regra Ri avaliando e agregando, por meio de uma t-norma, os graus de pertinência dos valores dos atributos de e1 nos conjuntos fuzzy determinados por Ri;
- 2. Para a mesma regra Ri repetir o passo anterior, calculando os graus de compatibilidade da regra Ri em relação a cada exemplo de entrada de e2 até ea:
- 3. O grau de cobertura da regra Ri em relação ao conjunto de dados ser a soma dos graus de compatibilidade de cada um dos exemplos de entrada em relação a regra Ri:

```
GdCRi = P \{GdC(Ri,e1), GdC(Ri,e2), ..., GdC(Ri,eq)\}
```

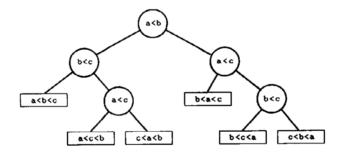
Uma vez calculados os graus de cobertura de todas as regras possíveis e definida a ordenação das regras em ordem decrescente desse grau, foram utilizados dois critérios diferentes para a pré-seleção de regras candidatas:

- 1. Considerar o conjunto de regras ordenadas até incluir todas as regras que apareceram na BC gerada pelo método de Wang & Mendel (WM) [15] e [16];
- 2. Considerar o conjunto de regras ordenadas contendo todas as regras com GdC não-nulo.

## 3.3 Fuzzy ID3(FID3)

O conceito de arvores de decisão foi popularizado por Quinlan [1] com o conceito do algoritmo Interactive Dichotomizer 3 (ID3).Um exemplo de uma arvore de decisão pode ser vista na figura 1 onde arvore conforme recebe a entrada toma uma decisão, classificando que letra possui maior valor que a outra:

Figura 1: Exemplo de arvore de decisão gerada utilizando ID3



Sistemas baseados nesta abordagem usam uma informação teórica, que é a medida de entropia, para avaliar o poder discriminatório de cada atributo. A característica mais importante das árvores de decisão é a sua capacidade de quebrar um problema complexo de tomada de decisão em um conjunto de decisões mais simples e, assim, proporcionar uma solução facilmente interpretável [17]. ID3 é um método eficiente de tomada de decisão para a classificação de dados simbólicos, sendo que geralmente, não é adequado em casos onde se deve operar valores numéricos, sendo que um outro problema comum do ID3 é que ele não pode fornecer qualquer informação sobre a região de intersecção onde os padrões de classes estão sobrepostos [18].

O algoritmo ID3 pode ser sintetizado da seguinte forma [19]:

Passo 1) calcular o valor de entropia [14].

Passo 2) escolher a característica no qual resulta na diminuição máxima na entropia (ganho de informação), para servir como o nó raiz da árvore de decisão.

Passo 3) construir o próximo nível da árvore de decisão fornecendo a maior diminuição na entropia.

Passo 4) repita os passos 1 a 3. Continue o procedimento até que todas as subpopulações sejam de uma única classe e a entropia do sistema igual a zero.

No ID3, um exemplo de treinamento em um conjunto particionado (e, portanto, em um nó) é binário. Contudo, no caso do Fuzzy ID3 (FID3), os conjuntos são nebulosos e, portanto, as ideias de conjuntos fuzzy e lógica fuzzy devem ser empregadas. Assim, com exceção desse caso, deve-se seguir as mesmas estratégias disponíveis para decisão simbólica árvores. Para modificar a forma em que o número de exemplos que são definidos para um nó é calculado, adapta-se normas utilizadas na lógica fuzzy para lidar com conjunções de proposições fuzzy [7].

O processo de particionamento (utilizado no FID3) cria uma árvore fuzzy, cunhando novas partições para os atributos que não existiam no particionamento definido anteriormente. O procedimento de particionamento pode ser detalhado em [8]. Árvores FID3 demonstram ser capazes de produzir conhecimentos tangíveis à compreensão, sendo ainda capazes de refinar detalhes - dependendo da inferência utilizada no momento.

## 3.4 FuzzyDT-C4.5

O algoritmo de indução de arvore de decisão C4.5 [2] foi criado utilizando como base o algoritmo ID3. O algoritmo de indução avalia o conteúdo de uma série de atributos e iterativamente constrói uma árvore a partir dos valores desses atributos com as folhas da árvore de decisão, sendo o valor desse atributo, o objetivo. Nesse procedimento, em cada passo de aprendizagem, a árvore evolutiva é ramificada no atributo que particiona os itens de dados de acordo com o maior ganho de informação. São adicionados ramos até que todos os itens no conjunto de treino sejam classificados. Para reduzir o efeito de overfitting, o C4.5 poda toda árvore de decisão construída. Ele recursivamente examina cada sub-árvore para determinar se substituindo-a por uma folha ou ramo reduziria erro previsão. Isto faz com que a poda na árvore de decisão seja a melhor em lidar com os dados a partir dos dados de formação [20].

Pulkkinen [9] aplica o algoritmo c4.5 no tratamento da população a ser utilizada pelo classificador fuzzy, e logo após gera o restante da população aleatoriamente, aplicando assim a logica fuzzy na arvore de decisão criada conforme elucidado em [9].

Uma versão baseada no C4.5, chamada FuzzyDT foi desenvolvida por [10] e explanada com mais detalhes em [11] e seu funcionamento é descrito como:

Passo1) definir a base de dados fuzzy, com a separação fuzzy para os domínios de características continuas.

Passo2) associar os atributos contínuos de treinamento com os rótulos linguísticos dos conjuntos fuzzy com maior compatibilidade com os valores de entrada.

Passo3) calcular a entropia e o ganho de informação para cada característica separada para dados de treinamento e definir os nós de testes da arvore até que todas as características sejam utilizadas ou todos os exemplos de treinamentos classificados

Passo4) aplicar o processo de pós-poda, similarmente ao C4.5 não fuzzy, usando por exemplo, 25% como limite de confiança como padrão.

#### 3.5 Fcart

O FCART é uma versão fuzzy do algoritmo de CART [21]. O algoritmo CART lida com atributos categóricos e contínuos para construir uma árvore de decisão, mesmo que apresente dados incompletos.

O CART usa o coeficiente de Gini [22] como medida de seleção de atributos para construir uma árvore de decisão. Ao contrário dos algoritmos ID3 e C4.5, o CART produz divisões binárias, por isso, produz árvores binárias. Sendo assim utilizando o coeficiente de Gini, não se utiliza premissas probabilísticas como ID3 e C4.5. O CART usa poda baseada no custo de complexidade para remover os ramos não confiáveis a partir da árvore de decisão para melhorar a precisão [23].

O CART constrói a árvore, dividindo de forma recursiva o espaço de variáveis baseada na impureza das variáveis para determinar a divisão até que a condição de encerramento seja encontrada. A impureza de gini determina quantas vezes um elemento escolhido aleatoriamente a partir do conjunto seria incorretamente marcado se fossem realizadas marcas aleatoriamente de acordo com a distribuição de rótulos no subconjunto [24].

Um exemplo da lógica fuzzy ao algoritmo CART padrão é utilizar em conjunto com o método de inferência ANFIS(Adaptative Neuro-fuzzy Inference Systems) mais detalhado em [25] e [26].

No ANFIS, tanto as capacidades de aprendizagem de uma rede neural e capacidades de raciocínio de lógica fuzzy foram combinadas a fim para se obter as capacidades de previsão melhoradas, quando comparado à utilização de uma metodologia única sozinha. O objetivo da aprendizagem ANFIS é encontrar um modelo ou mapeamento que irá associar corretamente os valores de entrada com os valores de objetivo. O sistema de inferência fuzzy (FIS) é uma representação do conhecimento, onde cada regra difusa descreve um comportamento local do sistema. Uma estrutura de rede que implementa FIS e emprega regras híbridas de aprendizagem necessárias para treinar é chamado ANFIS [27].

## 3.6 FlexDT

O algoritmo chamado de FlexDT por [12], é um classificador proposto para trabalhar com fluxo de dados que busca aprender de forma on-line, atendendo cada instância apenas uma vez, diferenciando o conceito de mudança do ruído e ainda, que visa ser aplicável no contexto impreciso, ou seja, lidar com a incerteza nos dados [12].

O Flexdt possui inúmeras vantagens. A primeira dela, ele adota aprendizagem incremental, sendo necessário processar cada instancia apenas uma vez. Uma segunda vantagem é que ele pode se adaptar devido a sua estrutura flexível a mudança de padrão efetivamente e eficientemente. Uma outra vantagem, que como herda a lógica Fuzzy, o Flexdt, é um modelo que lida bem com ruídos, sendo que uma quantidade moderada de ruídos não atrapalha o seu desempenho geral de classificação. Uma quarta vantagem é que ele é apto a manipular incerteza nos dados, sendo que se uma instancia chega com um valor ausente para qualquer atributo, essa instancia será pertencente a todos os conjuntos fuzzy desse atributo com os mesmos graus de pertinência. Dessa forma o FlexDT, é o primeiro algoritmo desenvolvido que sempre estende da lógica fuzzy de mérito para classificação de fluxo de dados [12].

Normalmente, quando falamos de particionamento de características fuzzy, temos como opções, o particionamento binário ou de múltiplo caminho. O Flexdt utiliza o particionamento binário fuzzy [13].

A partir deste modelo, [12] definiu as seguintes funções (2) e (3):

$$\mu_j L(x^i) = \frac{1}{\frac{x_j^i - \alpha_j}{\sigma_{jL}}}$$
(2)

$$\mu_j R(x^i) = \frac{1}{\frac{x_j^i - \alpha_j}{\sigma_{jR}}} \tag{3}$$

Onde  $\alpha_j$  e  $\sigma_j$  são o centro e a dispersão para cada função de pertinência sigmoidal respectivamente. O valor de  $\alpha_j$  é determinado de maneira similar a seleção do ponto de corte encontrado nas arvores de decisão tradicionalmente binária [13] .Para cada conjunto de instancias fuzzy S e característica  $x_j$ , a informação de entropia paras as partições direita e esquerda( $S_R$ ,  $S_L$ ) é estimado para qualquer provável ponto de corte  $\alpha$  da seguinte maneira em (4):

$$E\left(x_{j},S\right) = \frac{|S_{L}|}{S}I(S_{L}) + \frac{|S_{R}|}{S}I(S_{R}) \tag{4}$$

Onde (5):

$$I(S) = \sum_{k=1}^{q} P(c_k, S) Log(P(c_k, S))$$
(5)

Da classe  $c_k$  o ponto de corte  $\alpha_j$  é escolhido onde a entropia  $E(x_j, \alpha, S)$  é a menor entre todos as possíveis. Mantendo esse ponto de corte  $\alpha_j$ , e com os valores iniciais de  $\sigma_j$ .

Para os conjuntos fuzzy direita (R) e esquerda (L), são obtidos os desvios padrões das instancias, respectivamente. Os valores de  $\alpha_j$  e  $\sigma_j$  são atualizados durante a tarefa de aprendizagem, o que provê a flexibilidade de FlexDT em adaptar-se as mudanças de padrões [12]. A descrição completa do algoritmo pode ser encontrada em [12].

## 3.7 FuzzyDTC- Aplicação do gradiente ótimo no cálculo de cobertura da regra

Após revisarmos esses métodos existentes de arvores de decisão fuzzy na literatura citados, notamos cada um com sua peculiaridade possui uma maneira para lidar com alguns dados dependendo da distribuição desses dados. Visando reduzir esse impacto causado pela distribuição das amostras e de algumas possíveis incertezas já citadas, e inspirado pela fusão de métodos propostas em alguns casos na literatura como em [28] mesclando método Simplex e o de pontos interiores, este trabalho propõe a criação de um método generalista mesclando o núcleos de atualização e cálculo de pertinência do método FuzzyDT com as funções de pertinência do algoritmo FCart e FlexDT, e ainda utilizando a atualização online de cada nós da arvore proposta pelo FlexDT, dessa forma possibilitando uma generalização maior com tipos mais diferentes de distribuições, reduzindo a incerteza, utilizando da lógica de uma arvore baseada no c4.5 no FuzzyDT.

Inspirado pelos métodos de aproximação de pontos interiores que visa buscar um ponto no quase ótimo em [29] e [30], e que enxergam as projeções de separação como hiperplanos, este trabalho propõe na adequação do cálculo do grau cobertura das regras criadas enxergar as funções como hiperplanos da relação  $f(x_i) = \mu_a$ , encontrando assim o centro das interceções das funções visando encontrar um valor aceitável, que geralmente, tende a estar mais próximo da função com maior precisão, entretanto dentro do limite aceitável das outras, procurando reduzir o erro das funções de pertinência e ser mais flexível.

Yuan [31] define como uma forma de calcular a projeção do gradiente que é utilizada em [10] e [32] da seguinte forma:

$$H_{k+1} = H_k + \lambda_k \nabla_H^P F(W^t, H_k) \tag{6}$$

Ou também definido por [33] como

$$X_{k+1} = X_k + \alpha_k g_k \tag{7}$$

Onde podemos definir para encontrar  $\lambda_k$  ou  $\alpha_k$ :

$$\frac{\varepsilon(x_k) - \varepsilon(x_{k+1})}{\varepsilon(x_k)} \tag{8}$$

O método FuzzyDTC parte do algoritmo FuzzyDT como padrão para criação da arvore de decisão e em seu núcleo de atualização onde temos para cálculo da função de pertinência e comumente aplicamos uma função triangular, aplicamos também a função sigmoidal para cálculo da classe direita utilizada no Flexdt [12]

começando de um valor 0 em direção a 1 e também uma função senoidal comumente utilizada no algoritmo do fcart. A atualização do núcleo é descrito a seguir:

- 1-Ao obtermos a leitura do dado a ser trabalhado como fuzzy, aplicar x<sub>i</sub>na função triangular e guardar o resultado, juntamente com o cálculo da entropia [14] padrão S.
- 2-Aplicar a mesma entrada  $x_i$  na função de pertinência sigmoidal e e sua inversa do método Flexdt afim de se calcular o grau de adesão e não adesão e armazenar tais valores.
- 3- Aplicar a mesma entrada  $x_i$  na função de pertinência senoidal do método Fcart (anfis utilizando o gradiente) e sua inversa com base no método Flexdt afim de se calcular o grau de adesão e não adesão e armazenar tais valores.
- 4- Com posse dos dados calculados em 1,2,3 vamos agora vincular as funções 2 e 3 com a função triangular em 1 através do cálculo da relação da função 1 com a média da soma dos valores calculados em 2 e 3 através de (8) . Fazer o mesmo da função em 2 em relação a média dos valores obtidos e 1 e 3. Fazer o mesmo da função em 3 em relação a média dos valores obtidos e 1 e 2. Calcular o erro quadrático médio dos três erros encontrados.
- 5- Com posse das informações do erro obtidas no passo 4, incrementar o grau de pertinência calculado em 1 conforme a regra de incremento de pesos do gradiente ótimo em (6) e (7).

# 4 Experimento

O experimento consiste na análise comparativa entre a eficiência dos algoritmos de arvores de decisão fuzzy conforme expostos no trabalho. Para a realização dos mesmos, os algoritmos foram implementados e foram aplicados á testes com bases de dados da universidade da Califórnia (UCI), no total de 11 datasets conforme tabela I.

O experimento foi realizado sobre as mesmas circunstancias computacionais. Os códigos foram desenvolvidos em java baseados nos códigos disponibilizados por [11], e os dados coletados afim do estudo comparativo são a taxa do erro, e o desvio padrão do erro.

Cada algoritmo foi executado 5 vezes sequencialmente para cada base de dados, e os dados foram coletados via console do terminal java. Os testes de validação foram realizados utilizando o método de validação cruzada.

Com base na informação dos erros de predição obtidos foi realizado o teste anova para analisar a variância entre os erros, e realizar o teste de hipótese. O teste foi realizado com nível de confidencia  $\alpha = 0.1$  e com o poder do teste de  $\beta = 0.90$ , onde definimos Nosso H0 e H1:

$$h_0$$
:  $\mu_0 = 20$ 

$$h_1$$
:  $\mu_0 < 20$ 

Tabela 1: Detalhes das Bases Utilizadas

	Nº de instâncias	Nº de atributos	Nº de Classes	Desbalanceada
Vertebral Column	310	6	2	sim
Seeds	210	7	3	não
Iris	150	4	3	não
Sonar	208	60	2	não

Ionosphere	351	34	2	não
Ecoli	336	8	8	sim
Fertility	100	10	2	sim
cancer	699	10	2	sim
transfusion	748	5	2	sim
diabetes 130	100000	55	2	sim
car Evaluation	1728	6	4	sim

# 5 Resultados

Realizando o teste anova, para os níveis  $\propto = 0.1$ ,  $\beta = 0.90$ , variância entre grupos(between.var) = 20 e variância entre o mesmo grupo(within.var) = 40 obtivemos um número necessário de amostras para cada classe de 9, sendo assim as 11 bases analisadas para cada algoritmo são suficientes.

Os resultados são apresentados conforme tabela 2, 3, 4, 5 abaixo:

Tabela 2: Resultados FID3,FuzzyDT,fcart

	FI	D3	Fuzz	zyDT	Fca	art
Dataset	Erro	Desvio Padrão	Erro	Desvio Padrão	Erro	Desvio Padrão
Column	8,4005	0,0990	16,6666	0,0000	6,8667	0,0282
Seeds	21,0526	0,0000	15,7895	0,0000	21,0526	0,0000
Iris	8,9231	0,0240	7,6923	0,0000	10,1539	0,0346
Sonar	15,0000	0,0000	0,0000	0,0000	15,0000	0,0000
Ionosph ere	5,7059	0,0054	6,0588	0,0042	5,8235	0,0092
Ecoli	4,7619	0,0000	9,8095	0,0084	4,7619	0,0000
Fertility	11,1111	0,0000	11,1111	0,0000	11,1111	0,0000
cancer	1,0145	0,0062	1,4493	0,0000	0,9275	0,0054
Transfus ion	23,2877	0,0000	23,2877	0,0000	23,2877	0,0000
Diabetes	18,0790	0,0276	22,4211	0,0128	18,5005	0,0198
Car	20,1053	0,0300	19,8012	0,0296	20,7018	0,0268
Media	12,4947	0,0175	12,1897	0,0050	12,5625	0,0113

Tabela 3: Resultados FLEXDT, FUZZYDTC

	FlexDT			FUZZYDTC			
Ddataset	Erro	Desvio Padrão	Erro	Desvio Padrão			
Column	13,8000	0,0684	7,1333	0,0478			
Seeds	63,1579	0,0000	21,0526	0,0000			
Iris	38,4615	0,0000	7,6923	0,0000			
Sonar	15,0000	0,0000	15,0000	0,0000			
Ionosphere	14,7059	0,0036	6,0000	0,0024			
Ecoli	9,5238	0,0000	4,8571	0,0000			

Média	22,6008	0,0087	12,0392	0,0065
Car	22,8070	0,0000	6,0000	0,0024
Diabetes	23,7105	0,0232	15,0000	0,0000
transfusion	23,2877	0,0000	7,6923	0,0000
Câncer	13,0435	0,0000	21,0526	0,0000
Fertility	11,1111	0,0000	7,1333	0,0478

Figura 2: Comparação distribuição de Erros

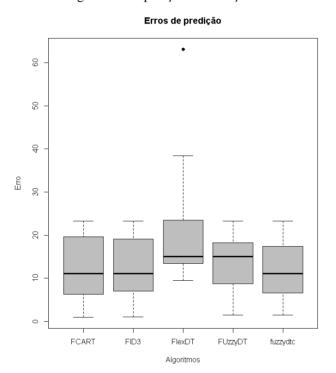


Tabela 4: Níveis de Especificidade x Sensibilidade: FID3, FUZZYDT, FCART

	F	TID3	Fuzz	yDT		Fcart
Dataset	Sensibilidade	Especificidade	Sensibilidade	Especificidade	Sensibilidade	Especificidade
Column	0,7667	0,7667	0,7667	0,9667	0,8667	0,8667
Seeds	0,8947	0,9474	0,8947	0,9474	0,7895	0,7895
Iris	0,9231	0,9615	0,9231	0,9615	0,9231	0,9615
Sonar	1,0000	1,0000	1,0000	1,0000	0,8500	0,8500
Ionosphere	0,9118	0,9118	0,9412	0,9412	0,9412	0,9412
Ecoli	0,9048	0,9048	0,9048	0,9048	0,9524	0,9524

Media	0,8829	0,9180	0,8835	0,9382	0,8754	0,9091
car	0,9064	0,9688	0,8830	0,9610	0,8363	0,9357
diabetes	0,7632	0,7632	0,7632	0,7632	0,8289	0,8289
transfusion	0,7671	1,0000	0,7671	1,0000	0,7671	1,0000
cancer	0,9855	0,9855	0,9855	0,9855	0,9855	0,9855
Fertility	0,8889	0,8889	0,8889	0,8889	0,8889	0,8889

Tabela 5: Níveis de especificidade x Sensibilidade: FLEXDT.FUZZYDTC

	Flex	DT	FUZZYDTC		
Dataset	Sensibilidade	Especificidade	Sensibilidade	Especificidade	
Column	0,9333	0,9333	0,7667	0,7667	
Seeds	0,7895	0,8947	0,8947	0,9474	
Iris	0,9231	0,9615	0,9231	0,9615	
Sonar	0,8500	0,8500	1,0000	1,0000	
Ionosphere	0,9412	0,9412	0,9412	0,9412	
Ecoli	0,9524	0,9524	0,8190	0,8190	
Fertility	0,8889	0,8889	0,8889	0,8889	
cancer	0,9855	0,9855	0,9855	0,9855	
transfusion	0,7671	1,0000	0,7671	1,0000	
diabetes	0,8289	0,8289	0,8158	0,8158	
car	0,8363	0,9454	0,9591	0,9864	
Media	0,8815	0,9256	0,8874	0,9193	

#### 6 Conclusões

Com o p-valor encontrado p=0.0637, sendo maior que o nível de significância ∝= 0.1, e o teste se trata de um teste uni-lateral a esquerda, devemos rejeitar H0, onde há evidência estatística para concluir que algoritmos de arvore decisão fuzzy, possuem uma comprovadamente uma taxa média de erro menor que 20 %;

Através desses resultados conseguimos perceber que arvores de decisão fuzzy podem ser aplicadas para classificação de classes com resultados positivos. Conseguimos notar ainda que o algoritmo proposto que modifica o algoritmo FuzzyDT para atualização online conforme semelhante ao FlexDT faz, aplicando o método do gradiente ótimo no cálculo do grau de cobertura obtém uma melhora visível, obtendo entre todos os métodos comparados a menor taxa de erros na predição, demonstrando assim a viabilidade de se utilizar o gradiente ótimo na otimização da generalização de um classificador.

Através da figura 2 é possível verificar que a distribuição dos erros do algoritmo fuzzydte é mais constante reduzindo assim o efeito da distribuição dos dados de entrada, sendo mais próxima de uma distribuição normal e alcançando a menor taxa de erros.

Analisando as tabelas 4 e 5, podemos verificar que o algoritmo FuzzyDTC apresenta o maior nível de Sensibilidade média, e um bom nível de especificidade média, próxima do algoritmo FUZYDT original.

Desta forma nota-se que algoritmos de arvores de decisão fuzzy se comportam muito bem como classificadores, trabalhando bem com várias bases diferentes, mantendo um comportamento confiável, e obtendo resultados expressivos.

Sugerimos como estudos futuros, a otimização da atualização online visando reduzir a taxa de erros, aprofundando linhas para otimização do espaço de busca dos menores erros.

## Referências

- [1] QUINLAN, J. R. Induction on decision trees. *Machine Learning*, Springer, v. 1, p. 81-106, 1986. Disponivel em: <a href="https://doi.org/10.1007/BF00116251">https://doi.org/10.1007/BF00116251</a>>. Acesso em: 30 set. 2017.
- [2] QUINLAN, J. R. Programs for Machine Learning. California: Morgan Publishers San Mated, 1993.
- [3] WANG, X.-Z.; DONG, L.-C.; YAN, J.-H. Maximum Ambiguity-Based Sample Selection in Fuzzy Decision Tree Induction. *Ieee Transactions On Knowledge And Data Engineering*, v. 24, n. 8 ,p. 1491-1505, 2012. Disponivel em: <a href="https://doi.org/10.1109/TKDE.2011.67">https://doi.org/10.1109/TKDE.2011.67</a>>. Acesso em: 30 set. 2017.
- [4] YUAN, Y.; SHAW, M. J. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, Elsevier, v. 69.2 ,p. 125-139, 1995. Disponivel em: <a href="https://doi.org/10.1016/0165-0114(94)00229-Z">https://doi.org/10.1016/0165-0114(94)00229-Z</a>. Acesso em: 30 set. 2017.
- [5] OLARU, C.; WEHENKEL, L. A complete fuzzy decision tree technique. *Fuzzy sets and systems*, Elsevier, v. 138.2 ,p. 221-254, 2003. Disponivel em: <a href="https://doi.org/10.1016/S0165-0114(03)00089-7">https://doi.org/10.1016/S0165-0114(03)00089-7</a>. Acesso em: 30 set. 2017.
- [6] ICHIHASHI, H. ;et al. Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning. *Fuzzy sets and systems*, Elsevier, v. 81 ,p. 157-167, 1996. Disponivel em: <a href="https://doi.org/10.1016/0165-0114(95)00247-2">https://doi.org/10.1016/0165-0114(95)00247-2</a>. Acesso em: 30 set. 2017.
- [7] JANIKOW, C. Z. Fuzzy decision trees: Issues and methods. *IEEE Trans. Syst., Man, Cybern.*, v. 28 ,p. 1-14, 1998. Disponivel em: <a href="https://doi.org/10.1109/3477.658573">https://doi.org/10.1109/3477.658573</a>. Acesso em: 30 set. 2017.
- [8] JANIKOW, C. Z. Fuzzy Partitioning with FID3.1. *18TH INTERNATIONAL CONFERENCE OF THE NORTH AMERICAN. IEEE*, p. 467-471, 1999. Disponivel em: <a href="https://doi.org/10.1109/NAFIPS.1999.781737">https://doi.org/10.1109/NAFIPS.1999.781737</a>. Acesso em: 30 set. 2017.
- [9] PULKKINEN, P.; KOIVISTO, H. Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms. *International Journal of Approximate Reasoning*, Elsevier, v. 48, n. 2, p. 526-543, 2008. Disponivel em: <a href="https://doi.org/10.1016/j.ijar.2007.10.004">https://doi.org/10.1016/j.ijar.2007.10.004</a>. Acesso em: 30 set. 2017.
- [10] CINTRA, M. E.; MONARD, M. C.; CAMARGO, H. A. FuzzyDT—a fuzzy decision tree algorithm based on C4. 5. *PROCEEDINGS OF THE BRAZILIAN CONGRESS ON FUZZY*, p. 199-211, 2012. Disponivel em: <a href="http://www.lbd.dcc.ufmg.br/colecoes/cbsf/2012/0016.pdf">http://www.lbd.dcc.ufmg.br/colecoes/cbsf/2012/0016.pdf</a>>. Acesso em: 30 set. 2017.
- [11] CINTRA, M.; MONARD, M. C.; CAMARGO, H. A. A fuzzy Decision Tree Algorithm based on c4.5. *Mathware* & *Soft computing*, v. 20, p. 56-62, 2013. Disponivel em: <a href="http://www.eusflat.org/msc/docs/vol20n1">http://www.eusflat.org/msc/docs/vol20n1</a> brasil4.pdf>. Acesso em: 30 set. 2017.
- [12] HASHEMI, S.; YANG, Y. Flexible decision tree for data stream classification in the presence of concept change, noise and missing values. *Data Mining and Knowledge Discovery*, Springer, v. 19, n.

- 1, p. 95-131, 2009. Disponivel em: <a href="https://doi.org/10.1007/s10618-009-0130-9">https://doi.org/10.1007/s10618-009-0130-9</a>. Acesso em: 30 set. 2017.
- [13] ISAZADEH, A.; MAHAN, F.; PEDRYCZ, W. MFlexDT: multi flexible fuzzy decision tree for data stream classification. *Soft Computing*, Springer, v. 20, n. 9 ,p. 1-15, 2015. Disponivel em: <a href="https://doi.org/10.1007/s00500-015-1733-2">https://doi.org/10.1007/s00500-015-1733-2</a>. Acesso em: 30 set. 2017.
- [14] SHANNON, C. E. Prediction and entropy of printed English. *Bell system technical journal*, v. 30, n. 1 ,p. 50-64, 1951. Disponivel em: <a href="http://dx.doi.org/10.1002/j.1538-7305.1951.tb01366.x">http://dx.doi.org/10.1002/j.1538-7305.1951.tb01366.x</a>. Acesso em: 30 set. 2017.
- [15] WANG, L. The WM method completed: a flexible fuzzy system approach to data. *IEEE trans. on Fuzzy Systems*, v. 11, p. 768-782, 2003. Disponivel em: <a href="https://doi.org/10.1109/TFUZZ.2003.819839">https://doi.org/10.1109/TFUZZ.2003.819839</a>>. Acesso em: 30 set. 2017.
- [16] WANG, L.; MENDEL, J. Generating fuzzy rules by learning from examples. *IEEE transactions on SMC*, v. 22, p. 414-427, 1992. Disponivel em: <a href="https://doi.org/10.1109/21.199466">https://doi.org/10.1109/21.199466</a>>. Acesso em: 30 set. 2017.
- [17] SAFAVIAN, S. R.; LANDGREBE, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst, Man, Cybern*, v. 21 ,p. 660-674, 1991. Disponivel em: <a href="https://doi.org/10.1109/21.97458">https://doi.org/10.1109/21.97458</a>. Acesso em: 30 set. 2017.
- [18] MITRA, S.; KONWAR, K. M.; PAL, S. K. Fuzzy Decision Tree, Liguistic Rules and Fuzzy Knowledge-Based Network: Generation and Evaluation. *IEEE transactions on systems, man, and cybernetics part c: Applications and Reviews*, v. 32, n. 4 ,p. 328-339, 2002. Disponivel em: <a href="https://doi.org/10.1109/TSMCC.2002.806060">https://doi.org/10.1109/TSMCC.2002.806060</a>>. Acesso em: 30 set. 2017.
- [19] PAO, Y.-H. Adaptive Pattern Recognition and Neural Networks. [S.l.]: Addison-Wesley Pub (Sd), 1989.
- [20] SORNLERTLAMVANICH, V.; POTIPITI, T.; CHAROENPORN, T. Automatic corpus-based Thai word extraction with the C4. 5 learning algorithm. *PROCEEDINGS OF THE 18TH CONFERENCE ON COMPUTATIONAL LINGUISTICS*, v. 2, p. 802-807, 2000. Disponivel em: <a href="https://doi.org/10.3115/992730.992762">https://doi.org/10.3115/992730.992762</a>>. Acesso em: 30 set. 2017.
- [21] BREIMAN; et al. Classification and Regression Trees. Mezzovico, Switzerland: Wadsworth, 1984.
- [22] LERMAN, R. I.; YITZHAKI, S. IMPROVING THE ACCURACY OF ESTIMATES OF GINI COEFFICIENTS. *Journal of Econometrics*, p. 43-47, 1989. Disponivel em: <a href="https://doi.org/10.1016/0304-4076(89)90074-2">https://doi.org/10.1016/0304-4076(89)90074-2</a>. Acesso em: 30 set. 2017.
- [23] LAVANYA, D.; RANI, U. K. Performance evaluation of decision tree classifiers on medical datasets. *International Journal of Computer Applications*, v. 26, n. 4, 2011. Disponivel em: <a href="https://pdfs.semanticscholar.org/00d8/a7088836830b216c501ab289a14bf5f4ecf2.pdf">https://pdfs.semanticscholar.org/00d8/a7088836830b216c501ab289a14bf5f4ecf2.pdf</a>>. Acesso em: 30 set. 2017.
- [24] SANTHANAM, T.; SUNDARAM, S. Application of CART Algorithm in Blood Donors Classification. *Journal of Computer Science*, p. 548-552, 2010. Disponivel em: <a href="http://dx.doi.org/10.3844/jcssp.2010.548.552">http://dx.doi.org/10.3844/jcssp.2010.548.552</a>>. Acesso em: 30 set. 2017.
- [25] JANG, J.-S. R. ANFIS: adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on 23.3* ,p. 665-685, 1993. Disponivel em:

- <a href="https://doi.org/10.1109/21.256541">https://doi.org/10.1109/21.256541</a>>. Acesso em: 30 set. 2017.
- [26] JANG, J.-S. R. Input selection for ANFIS learning. *PROCEEDINGS OF THE FIFTH IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS*, v. 2, p. 1493-1499, 1996. Disponivel em: <a href="https://doi.org/10.1109/FUZZY.1996.552396">https://doi.org/10.1109/FUZZY.1996.552396</a>>. Acesso em: 30 set. 2017.
- [27] YILMAZ, I.; KAYNAR, O. Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. *Expert Systems with Applications*, Elsevier, v. 38, n. 5 ,p. 5958-5966, 2011. Disponivel em: <a href="https://doi.org/10.1016/j.eswa.2010.11.027">https://doi.org/10.1016/j.eswa.2010.11.027</a>>. Acesso em: 30 set. 2017.
- [28] TIJSSEN, G. A.; SIERKSMA, G. Balinski Tucker Simplex tableaus: dimensions, Degeneracy degrees, and interior points of optimal faces. *Mathematical programming*, Springer, v. 81, n. 3, p. 349-372, 1998. Disponivel em: <a href="https://doi.org/10.1007/BF01580088">https://doi.org/10.1007/BF01580088</a>>. Acesso em: 30 set. 2017.
- [29] SHU, C.; CHEN, W. On Optimal Selection Of Interior Points For Applying Discretized Bounduary Conditions In Dq Vibration Analysis Of Beams And Plates. *Journal of sound and Vibration*, Elsevier ,p. 239-257, 1999. Disponivel em: <a href="https://doi.org/10.1006/jsvi.1998.2041">https://doi.org/10.1006/jsvi.1998.2041</a>>. Acesso em: 30 set. 2017.
- [30] AHARONI, R.; BERMAN, A.; CENSOR, Y. An interior points algorithm for the convex feasibility problem. *Advances In Applied Mathematics*, Elsevier, v. 4, n. 4 ,p. 479-489, 1983. Disponivel em: <a href="https://doi.org/10.1016/0196-8858(83)90019-2">https://doi.org/10.1016/0196-8858(83)90019-2</a>. Acesso em: 30 set. 2017.
- [31] YUAN, B. ;et al. NeNMF: An optimal Gradient Method for Nonnegative Matrix Factorization. *IEEE transactions on Signal Processing*, v. 60, n. 6, p. 2882-2898, 2012. Disponivel em: <a href="https://doi.org/10.1109/TSP.2012.2190406">https://doi.org/10.1109/TSP.2012.2190406</a>>. Acesso em: 30 set. 2017.
- [32] BUI, D. T.; et al. Hybrid artificial intelligence approach based on neural fuzzy inference model and metaheuristic optimization for flood susceptibility modeling in a high-frequency tropical cyclone area using GIS. *Journal of Hydrology*, n. 540, p. 317-330, 2016. Disponivel em: <a href="https://doi.org/10.1016/j.jhydrol.2016.06.027">https://doi.org/10.1016/j.jhydrol.2016.06.027</a>>. Acesso em: 30 set. 2017.
- [33] LI, J.; WANG, H. Maximum power point tracking of photovoltaic generation based on the optimal gradient method. 2009 Asia-Pacific Power and Energy Engineering Conference IEEE, p. 1-4, 2009. Disponivel em: <a href="https://doi.org/10.1109/APPEEC.2009.4918478">https://doi.org/10.1109/APPEEC.2009.4918478</a>. Acesso em: 30 set. 2017.