Árvores de Padrões Fuzzy Multi-Objetivo

Anderson Rodrigues dos Santos, Jorge Luis Machado do Amaral

Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rua São Francisco Xavier 524

Abstract. This paper describes a multi objective system for induction of fuzzy classifiers based on Fuzzy Pattern Trees. This is a tree-based hierarchical model, having as internal nodes, fuzzy logical operators and their leaves are composed by the combination of fuzzy terms and input attributes. A tree is created for each class present in the problem. Each tree will be a "logical class description" allowing the interpretation of the result. The construction method of the original trees was replaced by the Cartesian Genetic Programming, in order to better explore the search space. The search is guided by two objectives the accuracy in the classification and the interpretability, therefore the NSGA II multi-objective algorithm was used for search solutions that contemplate both. The proposed Classifier was compared with Random Forests, Support Vector Machines and K Nearest Neighbors in several databases of the UCI Machine Learning Repository presenting a competitive and interpretable result. Another comparison made was between the proposed model and the Fuzzy Pattern Trees originally proposed. The proposed model presented competitive results and smaller trees.

Resumo Este artigo descreve um sistema multi-objetivo para indução de classificadores utilizando Árvores de Padrões Fuzzy. Este é um modelo hierárquico baseado em árvores, tendo como nós internos, operadores lógicos fuzzy e suas folhas são compostas pela combinação de termos fuzzy e atributos de entrada. Uma árvore é criada para cada classe presente no problema. Cada árvore será uma "Descrição lógica da classe" permitindo a interpretação do resultado. Foi substituído o método original de construção das árvores pela Programação Genética Cartesiana, com o intuito de melhor explorar o espaço de buscas. A busca é orientada por dois objetivos a acurácia na classificação e a interpretabilidade, portanto foi utilizado o algoritmo multi-objetivo NSGA II para buscar soluções que contemplem ambos. O Classificador proposto foi comparado com Random Forests, Support Vector Machines e K Nearest Neighbors, em diversas bases de dados do UCI Machine Learning Repository apresentando um resultado competitivo e interpretável. Outra comparação feita, foi entre o modelo de proposto e as Árvores de Padrões Fuzzy propostas originalmente. O modelo proposto apresentou resultados competitivos e árvores menores.

1 Introdução

A análise de uma grande quantidade de dados é um tarefa complexa e dependendo da quantidade não é possível ser feita por seres humanos. Portanto, esta tarefa cria um interesse considerável no estudo de modelos que podem aprender a partir de um conjunto de dados [15]. A indução destes modelos pode ser feita de uma maneira automática, através da utilização de diferentes métodos, tais como: Redes Neurais Artificiais, Métodos Bayesianos, Modelos Gráficos, Árvores de Decisão, entre outros. Muitos desses métodos são chamados de "Black Box"pois por mais que o modelo gerado por eles apresente um excelente resultado, o modelo não é interpretável, não se sabe "como"o modelo induzido tomou sua decisão. Para se ter um conhecimento de "como", modelos gerados por abordagens simbólicas, por exemplo, sistemas fuzzy baseados em regras podem ser utilizados, pois geram modelos interpretáveis.

Alguns dos métodos com aplicações mais bem sucedidas na síntese de modelos interpretáveis são baseados na teoria dos conjuntos Fuzzy[2][6]. Pois sistemas fuzzy criam uma interface entre padrões quantitativos e estruturas de conhecimento qualitativas, expressas em linguagem natural. Esta característica faz com que sistemas fuzzy sejam atraentes do ponto de vista da representação do conhecimento, permitindo que o conhecimento adquirido a partir de um banco de dados possa ser representado de forma compreensível. Como resultado, ele dá ao modelo um estágio superior de interpretabilidade [7].

Este artigo descreve um sistema para a indução automática de modelos, aplicados a tarefa de classificação. O modelo utilizado se chama Árvores de Padrões Fuzzy (APF). APF é um modelo hierárquico baseado em árvores, tendo como nós internos, operadores lógicos fuzzy e as folhas das árvores são compostas por uma combinação de termos fuzzy e atributos de entrada. Neste modelo, o classificador é obtido criando-se uma árvore para cada classe. Cada árvore será uma "descrição logica da classe", permitindo a interpretação dos resultados. A estratégia para o aprendizado das árvores, proposta em [12] chamada "Beam Search", possui algumas limitações: A primeira está relacionada à característica gananciosa do algoritmo de busca, sempre procurando o melhor candidato no atual estágio da construção da árvore. Esta característica pode estreitar o espaço de busca, fazendo com que o algoritmo fique preso a uma solução sub ótima. Outra desvantagem é associada com a "Maldição da dimensionalidade". Se a quantidade de atributos e a largura do feixe forem grandes, o algoritmo levará muito tempo para avaliar todas as possibilidades, por esta razão, haverá uma explosão na quantidade de combinações possíveis. Por outro lado, se a largura do feixe é pequena, apenas uma pequena região do espaço de busca será explorada. Consequentemente limitando o algoritmo na tarefa de encontrar boas soluções. Para minimizar estes problemas, foi substituído o método de aprendizado utilizado anteriormente, pela Programação Genética Cartesiana(PGC), pois a PGC é um método de busca global muito eficiente que pode descobrir APF precisas e interpretáveis. A PGC é uma forma de programação genética, muito eficiente e flexível, que codifica um programa de computador em uma representação em grafos. Assim, esse grafos são utilizados para representar várias estruturas com-

putacionais, e podem ser usados facilmente para representar APFs. Além disso, a PGC é um método de pesquisa global capaz de explorar grandes espaços de busca eficientemente, buscando encontrar uma representação adequada de uma APF. A classificação feita através das árvores obtidas pela PGC devem apresentar uma boa acurácia , com árvores pequenas e interpretáveis. Outra alteração do método original é o emprego de operadores fuzzy de concentração e diluição. Estes operadores podem melhorar a representação linguística fornecida pela APF.

Os objetivos da abordagem proposta são: Apresentar um modelo que seja competitivo do ponto de vista da acurácia, conseguindo classificar os dados tão bem quanto outros métodos consagrados e criar modelos que sejam interpretáveis, o que neste caso seriam APF com árvores enxutas. Para conciliar e obter soluções que contemplem estes dois objetivos foi utilizado o algoritmo multi objetivo NSGA II.

Este artigo é dividido em sete seções. Na segunda seção será apresentada uma descrição resumida das Árvores de Padrões Fuzzy. Na terceira será apresentada uma descrição concisa da PGC. Na quarta é apresentado o algoritmo multi objetivo NSGA II Na quinta é descrito o método proposto e na sexta os resultados obtidos. Na sétima seção foram apresentadas as conclusões do trabalho.

2 Árvores de Padrões Fuzzy

As Árvores de Padrões Fuzzy foram criadas focando a representação do conhecimento através de uma expressão em forma de árvore ao invés de representa-lo em forma de regras. O primeiro método de indução de APF foi criado por Huang, Gedeon e Nikravesh [8], e mais tarde, o algoritmo de geração das árvores foi refinado em [12]. Esta representação hierárquica minimiza problemas existentes em sistemas baseados em regras, como o aumento exponencial da quantidade de regras com o aumento das entradas e a perda da interpretabilidade quando uma grande quantidade de regras for necessária para alcançar os requerimentos de acurácia. A árvore é representada como um grafo, favorecendo a habilidade humana de reconhecer padrões visuais, permitindo a descoberta de relações entre os atributos de entrada. Esta relação pode ser difícil de ser feita quando são utilizados modelos com um conjunto fixo de regras. As APF fornecem uma alternativa para a construção de modelos fuzzy precisos e interpretáveis. APF é um modelo hierárquico com uma estrutura de árvore, no qual os nós internos são operadores usados em sistemas fuzzy e as folhas são compostas por termos fuzzy associados com um atributo de entrada. No decurso da sua avaliação, a APF propaga a informação do fundo para o topo da árvore. Assim, os nós internos recebem valores de seus predecessores e os combina usando um operador. A saída da operação é apresentada ao nível superior da árvore. Consequentemente, a APF realiza um mapeamento recursivo, gerando a saída em um intervalo unitário. Um classificador baseado em árvores de padrões é desenvolvido pela criação de uma árvore para cada classe. A decisão do classificador ocorre em favor da árvore (classe) que possui o maior valor de saída. Também, desde que cada árvore é considerada

uma "descrição lógica" da classe, ela permite uma interpretação mais especifica do problema de aprendizado[12]. Um exemplo de APF pode ser visto na Figura 1, a árvore representa a classe "Vinho de boa qualidade". Os atributos de entrada são o teor alcoólico, a acidez, a concentração de dióxido de enxofre e sulfatos. Eles são relacionados a termos fuzzy que representam um intervalo do universo de discurso de um atributo. Por exemplo, na Figura 1, o termo fuzzy Álcool baixo, representa o conjunto fuzzy que indica um baixo teor alcoólico. Os valores de pertinência obtido nos conjuntos fuzzy são combinados por operadores que mantem os resultados parciais dentro do intervalo [0,1]. O valor obtido na saída após todas as combinações de características, deve se aproximar de 1, caso as características que compõe a árvore sejam úteis para a definição da classe que ela representa.

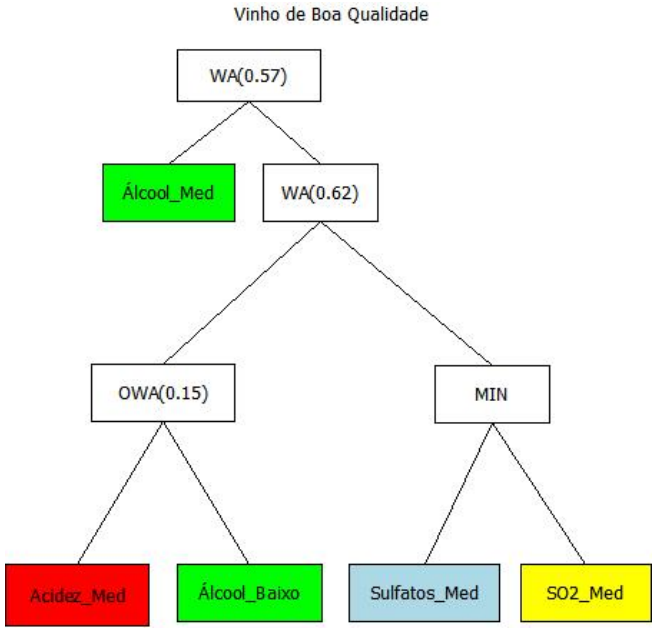

Figura 1. Árvore que representa a classe Vinho de boa qualidade. Foi atribuída uma cor diferente a cada atributo utilizado.

A associação entre um atributo e um termo fuzzy é representada por uma função de pertinência. O modelo baseado em árvores mapeia diversas entradas em somente 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 critérios e agregando-os posteriormente[12]. Observando a Figura 1, verifica-se que a qualidade do vinho está relacionada com duas sub-árvores. A primeira relaciona o teor alcoólico e a acidez, enquanto a outra relaciona a concentração de sulfatos e de dióxido de enxofre. As informações destas sub-árvores

são posteriormente combinadas em um nível mais alto. Neste tipo de sistema é possível identificar que as sub-árvores representam diferentes conhecimentos que devem ser combinados. Também é relevante notar que a análise dessas sub-árvores pode fornecer informações suplementares. Dependendo da configuração da árvore, se um atributo está mais próximo do topo ou do fundo, pode-se identificar quais atributos contribuem mais para o resultado final. Além dos termos fuzzy, t-norms, t-conorms, operadores de média e operadores de diluição, concentração e complemento foram usados na criação das árvores[12].

3 Programação Genética Cartesiana

A PGC [11] é uma variação de programação genética no qual os programas são representados por grafos. O grafo é codificado em uma sequência linear de inteiros representados em uma grade de nós computacionais. Embora, a grade possa ter qualquer número de dimensões, na maior parte das aplicações, ela apresenta somente uma ou duas. A PGC apresenta várias vantagens, tais como: Neutralidade, redundância e a ausência de um problema chamado "Bloat" (crescimento do programa sem retorno significativo em termos de aptidão), comum em outros métodos de programação genética [10] [9]. Na PGC, os grafos são representados em uma sequência linear de inteiros, que são chamados de cromossomos ou genótipos, como mostrado na Figura 2.

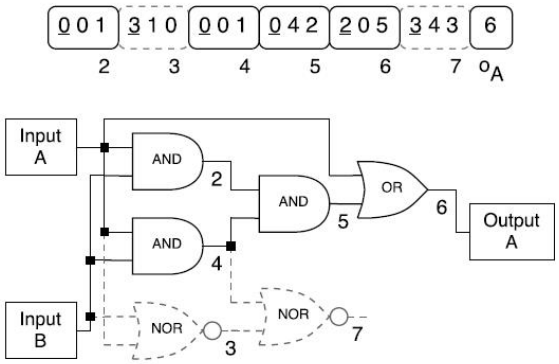

Figura 2. Genótipo e seu respectivo fenótipo

Esses cromossomos podem ser divididos em partes. Estas partes são chamadas de genes. Os genes podem ser rotulados como gene de função, conexão ou de saída. Um nó é formado por um gene de função e dois genes de conexão. A função (Rotulada pelo gene de função) usa como parâmetros de entrada os valores indicados pelos genes de conexão para gerar a saída do nó, que por sua vez pode ser utilizada como parâmetro de entrada de uma função em outro nó. Por

uma questão de desenvolvimento do processo evolutivo nos genótipos, é necessário decodificá-los em fenótipos para se obter a solução no domínio do problema. Quando o genótipo é decodificado, alguns nós podem não estar conectados ao fluxo de dados, criando um efeito de neutralidade [11]. Um exemplo pode ser visto na Figura 2. Enquanto o genótipo tem um tamanho fixo, os fenótipos podem variar. O operador mutação é utilizado na PGC, sendo frequentemente utilizado o "one-point mutation". A quantidade de genes que sofrem mutação é definida por uma porcentagem do total de genes, chamada de taxa de mutação. Nas Figuras 2 e 3 são mostrados o antes e depois de uma operação de mutação. Neste exemplo o último gene foi alterado; Pode-se ver que a mudança em um único gene pode transformar significativamente o fenótipo [11]. O algoritmo evolucionário usado na PGC é a estratégia evolucionária $1+\lambda$, onde o valor de λ frequentemente usado é quatro [11]. Essa estratégia permite a PGC encontrar boas soluções de forma eficiente e em poucas avaliações. A próxima sessão descreve como a PGC pode ser usada para sintetizar APF.

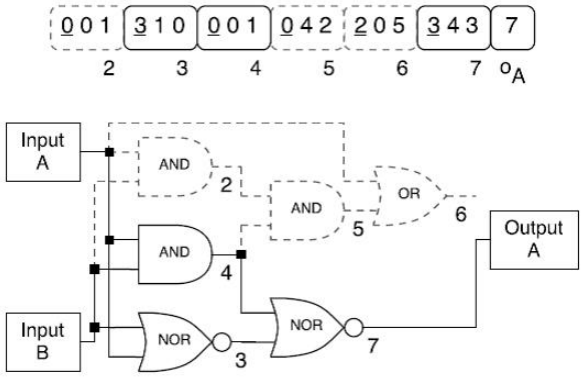

Figura 3. Genótipo-Fenótipo apos a mutação de um gene.

3.1 Melhorias CGP

Em [5] foi estudado como os operadores de busca interagem com a representação da solução na PGC. Os focos desta pesquisa foram a criação de métodos que evitem o desperdício de avaliações e a criação de métodos que superem a limitação do processo de busca imposta pelo ordenação do cromossomo. Foram propostas algumas melhorias interessantes para o processo de busca da PGC que foram utilizadas no modelo proposto neste artigo.

Otimização das avaliações O operador de mutação geralmente é aplicado aos genes ativos e inativos uniformemente. Na mutação feita desta forma clássica,

existe a possibilidade de se fazer mutação apenas em nós inativos. A prole pode ter os nós ativos idênticos aos ancestrais e consequentemente uma aptidão igual aos mesmos. Desta forma avaliar este descendente é um desperdício pois já é sabida a aptidão do mesmo[5] .

O método SINGLE foi proposto para evitar avaliações duplicadas. Neste método a mutação é modificada para garantir que apenas sejam criados descendentes cujos nós ativos não sejam idênticos ao ancestral. Para alcançar este objetivo, ao invés da mutação de cada gene a uma certa probabilidade, são modificados genes individuais aleatoriamente até que algum nó ativo seja modificado. Este método limita a deriva genética pois o gene ativo que sofreu a mutação deve ser um Intron ou outra forma de representar uma solução de qualidade igual. Outra vantagem deste método é evitar uma sobrecarga computacional por gerar repetidamente indivíduos que não são avaliáveis. Uma terceira vantagem deste método é que ele não requer um parâmetro de mutação [5].

Ordem do Cromossomo A PGC ordena os nós do cromossomo para evitar ciclos, garantindo que cada nó receba na entrada informações vindas de nós que o precedam. Esta restrição não limita a capacidade da PGC em representar grafos acíclicos dirigidos, pois todos os grafos acíclicos dirigidos podem ser serializados para se ajustar a este requerimento. Porem esta representação pode ter algum impacto na habilidade da PGC em evoluir grafos acíclicos dirigidos específicos [5].

Forçar a ordenação dos nós adiciona limitações artificiais a habilidade da PGC de conectar nós. Nós que poderiam ser conectados sem criar um ciclo podem não ser formados devido a ordenação aleatória do cromossomo. Similarmente, é impossível dois nós adjacentes se conectarem através de um nó intermediário. Consequentemente, estrutura uteis do cromossomo podem ter que evoluir diversas vezes para encontrar o melhor local para esta estrutura dentro do cromossomo [5].

Para corrigir essas questões e examinar a PGC sem viés de ordenação, foi criado o método REORDER . O método REORDER ordena os nós em um cromossomo existente sem modificar o comportamento dos nós. Isto é possível pois para um dado nó, podem existir uma grande quantidade de outros nós que não requerem uma ordem especifica, pois eles não tem suas entradas ou saídas ligadas direta ou indiretamente ao nó em questão. Neste método é designada uma nova locação para um nó no cromossomo aleatoriamente assim que todos os nós que estão conectados ao nó em questão foram realocados no cromossomo anteriormente [5].

4 NSGA II

O NSGA II é um algoritmo utilizado em otimização multi objetivo. A presença de múltiplos objetivos em um problema dá origem a um conjunto de soluções ao invés de uma única solução ótima. Nenhuma das soluções deste conjunto pode ser dita melhor que outra. Este algoritmo é popular e esta popularidade de acordo

com[3] é atribuída a sua baixa complexidade de computação e armazenamento, seu uso de elitismo e sua abordagem sem parâmetros.

O NSGA II utiliza um mecanismo chamado distancia de agrupamento. Esta métrica dá ao NSGA uma estimação de densidade para qualquer uma das fronteiras de Pareto. Ela é definida como a distancia entre duas soluções vizinhas em ambos os lados da solução ao longo do eixo de cada objetivo [4].

O problema de otimização multi objetivo pode ser definido pela minimização de:

$$f(x) = [f_1(x), f_2(x), ..., f_M(x)] \tag{1}$$

Onde M é o numero de funções objetivo $f_i : \mathbb{R}^n \longmapsto \mathbb{R}$. As soluções são compostas de n variáveis de decisão $x = [x_1, x_2, ..., x_n]$. Uma solução p domina outra solução q se $f_i(p) \leq f_i(q)$ para todo i=1,...,M e $f_j(p) < f_j(q)$ para ao menos um j. Esta relação de dominância pode ser usada para atribuir um ranque: Se um individuo tem Ranque 1 este indivíduo não é dominado, se o indivíduo está no ranque 2, este indivíduo é dominado por algum outro individuo do ranque 1. O ranque indica o índice da fronteira de Pareto ao qual a solução está associada. Para guiar a seleção no processo de evolução, o NSGA II define um operador chamado *crowded-comparison*($\prec_c$). Cada individuo i na população tem dois atributos: Um ranque de não dominação(i_{ranque}) e a distancia de agrupamento para o vizinho mais próximo (i_{dist}) [4]. A ordem criada pelo operador baseada nesses dois atributos e definida por:

$$i \prec_c j := i_{ranque} < j_{ranque} \vee (i_{ranque} = j_{ranque} \wedge i_{dist} > j_{dist}) \tag{2}$$

O Operador favorece soluções com um menor ranque de dominância quando as duas soluções estão em fronteiras diferentes e quando as soluções estão na mesma fronteira, a solução em uma região menos populada é favorecida.

5 Método Proposto

O método proposto utiliza a Programação Genética Cartesiana na construção das Árvores de Padrões Fuzzy. As vantagens de usar a PGC nesta tarefa são: A PGC tem uma representação em forma de grafos flexível, sendo facilmente adaptada para a representação das APF. A PGC busca eficientemente soluções dentro de um espaço de busca grande, fazendo com que o processo de construção das árvores seja menos sensível a maldição da dimensionalidade. Uma vez que a exploração do espaço de busca na PGC não depende da estratégia gananciosa do Beam Search, ela tem uma chance maior de conseguir soluções melhores. Além disso, não se restringe, como o Beam Search (que é limitado pela largura do feixe). Isso também aumenta as chances de obtenção de soluções melhores. O modelo proposto utiliza um algoritmo evolutivo, portanto seu êxito depende de como a solução é codificada e da escolha da função de aptidão para a avaliação da solução. As próximas duas subseções explicam cada uma dessas implementações.

5.1 Representação

Cada Atributo é rotulado por um de cinco termos linguísticos (termos fuzzy), os termos são, baixo, médio-baixo, médio, médio-alto e alto. Os termos linguísticos são obtidos pela partição uniforme do espaço dos atributos de entrada. Os genes de função podem representar os diferentes operadores que podem ser utilizados nas APF.Modificadores e complemento também foram implementados, uma vez que podem fornecer um significado linguístico adicional. Foram utilizados os seguintes operadores:

$$Maximo = MAX(a, b) \tag{3}$$

$$Minimo - MIN(a, b) \tag{4}$$

$$WA(k) = ka + (1 - k)b \tag{5}$$

$$OWA(k) = kMAX(a, b) + (1 - k)MIN(a, b) \tag{6}$$

$$Concentrador = a^2 \tag{7}$$

$$Diluidor = a^{\frac{1}{2}} \tag{8}$$

$$Complemento = 1 - a \tag{9}$$

Onde a e b são os valores das entradas dos nós que serão operados. No caso dos operadores WA e OWA, k será um valor criado de forma aleatória dentro do intervalo [0,1]. No caso do concentrador, diluidor e complemento a entrada b será ignorada.

A fim de obter o genótipo que irá representar a árvore, a PGC tem ao seu dispor todos os operadores e termos fuzzy. Dependendo do gene conexão utilizado como entrada do gene de função, diferentes árvores podem ser obtidas.

5.2 Função de Aptidão

A função de aptidão do objetivo que busca uma maior acurácia de classificação é calculada através da distancia entre o valor obtido na árvore que representa a classe de cada instancia da base de dados do treino e o maior valor obtido entre as árvores das outras classes. Realçando as diferenças entre as classes.

A função de aptidão do objetivo que busca a interpretabilidade, é calculada pela quantidade de genes ativos. Quanto menos genes ativos, menor a árvore resultante e portanto mais interpretável.

6 Resultados

Foram realizados alguns experimentos para avaliar o desempenho do modelo proposto em diversas bases de dados. Estas bases foram obtidas no UCI machine learning repository. Elas também são usadas em[12], portanto, é possível comparar as APF criadas pelo método proposto com as criadas em[12]. Adicionalmente, esses estudos fornecem uma comparação com outros classificadores bem conhecidos: Support Vector Machine com Kernel Linear (SVM-L)[13], K-nearest neighbors(KNN)[14], Random Forest (RF) [1] e Support Vector Machine com Kernel Radial Basis Function(SVM-R)[13]. O critério confrontado foi a estimativa da medida de desempenho de generalização realizada através de uma validação cruzada com 10 pastas e 5 repetições. Os parâmetros dos outros classificadores foram definidos da seguinte forma: O número de vizinhos do KNN foi fixado em 1; a quantidade de árvores geradas no Random Forest foi fixada em 50 e a quantidade de atributos sob o qual é feita a partição é igual a 1.

Os parâmetros do SVM foram determinados utilizando validação cruzada. Uma vez que os parâmetros ótimos foram encontrados, o classificador foi treinado com o conjunto de treinamento e seu desempenho foi avaliado no conjunto de teste. O procedimento usado para encontrar os parâmetros não foi tendencioso, pois a pasta de teste na validação não foi utilizada para ajustar os parâmetros. Apesar da validação cruzada requerer um maior custo computacional, ela fornece uma medida da habilidade de generalização, pois as pastas de teste não foram utilizadas para ajustar os parâmetros; Além disso, também ajuda a reduzir o *overfit*.

A tabela 1 apresenta a acurácia preditiva dos algoritmos . Nessa tabela, a medida de desempenho foi arredondada para duas casas decimais. A análise da tabela mostra que o APF-PGC MO apresenta resultados competitivos. Ele teve acurácia igual ou maior que 90 porcento em 3 bases de dados e uma diferença absoluta da acurácia para o melhor algoritmo de menos que 5 porcento em 6 bases de dados. A tabela 1 também apresenta o PTTDE.25(APF Original), que é o modelo com o melhor resultado de acurácia nas implementações anteriores de APF[12]. Esses resultados foram obtidos da implementação em java do algoritmo "Top-Down" fornecido pelos autores[12].

Tabela 1. Acurácia

Dataset	SVM-L	KNN	RF	SVM-R	PTTDE.25	APF-PGC MO
Iris	**0.96**	0.94	0.95	0.95	0.95	0.95
Lawsuit	**0.99**	0.96	0.97	0.96	0.94	0.93
Pima	**0.77**	0.70	**0.77**	0.71	0.76	0.72
Lupus	0.74	0.68	0.62	0.73	**0.77**	0.74
Transfusion	0.76	0.71	0.70	0.73	**0.77**	0.76
Wine	**0.98**	0.95	**0.98**	**0.98**	**0.98**	0.90
Haberman	0.72	0.67	0.65	0.71	**0.74**	0.73
Australian	**0.86**	0.80	0.79	0.85	0.85	0.85

Tabela 2. Profundidade média das Árvore

Dataset	PTTDE.25	APF-PGC(MO)
Iris	3.33	**1.24**
Lawsuit	9.00	**1.05**
Pima	4.00	**1**
Lupus	4.00	**1.65**
Transfusion	5.00	**2.00**
Wine	10.0	**1.00**
Haberman	2.00	**1.85**
Australian	5.00	**0.5**

Para validar a competitividade do APF-PGC MO, foram aplicados os testes de Friedman e o Nemenyi Post-Hoc. Foi comparado o desempenho dos classificadores e para verificar se é possível identificar um classificador que apresente um desempenho melhor, que seja estatisticamente significante (nível de significância dado por p-value=0.05) quando se leva em consideração todas as diferentes bases de dados dos experimentos. Os resultados determinaram que não há diferença estatisticamente significativa entre os classificadores. Também foi comparada a profundidade média das árvores APF-PGC MO e PTTDE.25. Dos resultados apresentados na tabela 2, pode-se concluir que a APF-PGC MO teve um melhor desempenho que o PTTDE.25 com respeito a profundidade média das árvores. Os resultados indicam que o método de busca "Beam-search" utilizado no PTTDE.25, tem um desempenho inferior, devido a sua característica gananciosa, pois ele escolhe as melhores sub-árvores candidatas no momento, sem voltar atrás, descartando candidatos que podem não ter um bom desempenho individualmente, mas quando combinados com outra sub-árvore escolhida posteriormente poderiam ter um desempenho melhor.

7 Conclusão

Este artigo propõe uma nova forma de indução de Árvores de Padrões Fuzzy utilizando Programação Genética Cartesiana como algoritmo de aprendizado em um ambiente multi-objetivo, utilizando o NSGA II. Os resultados obtidos demostraram que APF-PGC MO tem um desempenho competitivo na tarefa de classificação em relação a alguns dos melhores classificadores disponíveis, mas fornecendo um modelo interpretável, ou seja, o conhecimento obtido no aprendizado pode ser extraído do modelo. A abordagem multi objetivo e as diversas mudanças no algoritmo da PGC clássica contribuíram para uma busca mais eficiente, sem avaliações desperdiçadas e árvores mais enxutas. APF constituem uma alternativa viável aos clássicos modelos fuzzy baseados em regras, pois sua estrutura hierárquica permite uma representação mais compacta e um compromisso entre a acurácia e a simplicidade do modelo.

Agradecimentos

Agradecemos a Faperj pelo apoio e financiamento desta pesquisa.

Referências

1. Leo Breiman. Random forests. 45(1):5–32, 2001.
2. Oscar Cordón. A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. 52(6):894–913, 2011.
3. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. 6(2):182–197, 2002.
4. Félix-Antoine Fortin and Marc Parizeau. Revisiting the NSGA-II crowding-distance computation. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, pages 623–630. ACM, 2013.
5. B. W. Goldman and W. F. Punch. Analysis of cartesian genetic programming; evolutionary mechanisms. 19(3):359–373, 2015.
6. Francisco Herrera. Genetic fuzzy systems: taxonomy, current research trends and prospects. 1(1):27–46, 2008.
7. Eyke Hüllermeier. Fuzzy methods in machine learning and data mining: Status and prospects. 156(3):387–406, 2005.
8. Zhiheng Huang, T.D. Gedeon, and M. Nikravesh. Pattern trees induction: A new machine learning method. 16(4):958–970, 2008.
9. J.F. Miller and S.L. Smith. Redundancy and computational efficiency in cartesian genetic programming. 10(2):167–174, 2006.
10. Julian Miller. What bloat? cartesian genetic programming on boolean problems. In *2001 GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE LATE BREAKING PAPERS*, pages 295–302, 2001.
11. Julian F. Miller. *Cartesian Genetic Programming*. Springer, 2011 edition edition, 2011.
12. R. Senge and Eyke Hüllermeier. Top-down induction of fuzzy pattern trees. 19(2):241–252, 2011.
13. Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2nd edition edition, 1999.
14. Andrew R. Webb. Density estimation – nonparametric. In *Statistical Pattern Recognition*, pages 81–122. John Wiley & Sons, Ltd, 2002.
15. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., 2005.