https://eventos.utfpr.edu.br//sicite/sicite2020

Seleção de atributos utilizando método *wrapper*, rodadas e votos

Feature selection using wrapper method, rounds and votes

Jaqueline Sayuri Machida
jaquelinemachida@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Danilo Sipoli Sanches
danilosanches@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

RESUMO

Com o aumento da disponibilidade de dados, surgem também mais possibilidades de emprego do aprendizado de máquina. No entanto, muitos dados gerados não são desejáveis para o contexto do problema, e a seleção de atributos procura justamente eliminar os dados que prejudicam a performance de classificadores. Recentemente foi proposto em um trabalho um algoritmo *filter* de seleção de atributos, onde a seleção ocorre por **rodadas** e é introduzida uma etapa de **votação**, responsável por gerar subconjuntos de características. O presente trabalho buscou investigar o desempenho do algoritmo quando convertido para *wrapper,* visto que o método costuma atingir soluções que levam à melhores acurácias, sendo utilizado o Algoritmo Genético para explorar as combinações possíveis de atributos.

PALAVRAS-CHAVE: Seleção de atributos. *Wrapper*. Algoritmo genético.

ABSTRACT

With the increase in data availability, there are also more possibilities for using machine learning. However, much of the data generated is not desirable in the context of the problem, and the selection of attributes seeks to precisely eliminate the data that hinder the performance of classifiers. Recently, a filter algorithm for attribute selection was proposed where the selection takes place by **rounds** and a **voting** stage is introduced, responsible for generating subsets of characteristics. The present work sought to investigate the performance of the algorithm when converted to wrapper, since the method usually reaches solutions that lead to better accuracy, using the Genetic Algorithm to explore possible combinations of attributes.

KEYWORDS: Feature Selection. Wrapper. Genetic Algorithm.

Recebido: 19 ago. 2020.
Aprovado: 01 out. 2020.

INTRODUÇÃO

Com o aumento da disponibilidade de dados, surgem também mais possibilidades de emprego do aprendizado de máquina (*machine learning*). Toda essa informação permite que problemas das mais diversas áreas sejam resolvidos de forma rápida e eficiente pelo uso de algoritmos inteligentes. Como exemplos, pode-se citar o desenvolvimento de sistemas de recomendação, detecção de fraudes e predição de doenças. No entanto, muitos dados gerados não são desejáveis para o contexto do problema, pois podem ser irrelevantes e interferirem no desempenho de algoritmos classificadores, devido à correlação que pode existir entre esses dados e as classes, sem que de fato exista uma relação causal. Além dos dados irrelevantes, pode-se ressaltar também aqueles que são redundantes, de forma a aumentar o custo computacional sem trazer ganho na classificação.

A seleção de atributos/características (*feature selection*) procura justamente eliminar os dados que prejudicam a performance de classificadores. Os conjuntos de dados (*datasets*) utilizados no treinamento e teste dos algoritmos são tabelas, onde os dados são organizados em colunas (por exemplo: dados como idade e altura seriam armazenados em colunas diferentes por se tratarem de dados diferentes). Cada coluna representa um atributo/característica (no exemplo anterior, idade e altura são atributos/características). Assim, a seleção de atributos, como o próprio nome diz, busca melhorar a qualidade dos *datasets* selecionando atributos relevantes, não redundantes e sem ruídos, com o objetivo de aprimorar a classificação ou, pelo menos, mantê-la.

Segundo Mafarjha (2008), dado um conjunto de atributos, seus subconjuntos possíveis podem ser obtidos de forma exaustiva (cujo custo e tempo computacional aumenta quanto maior o conjunto), aleatória (pode acabar executando uma busca exaustiva) ou heurística (capaz de encontrar soluções aceitáveis em um tempo razoável). Entre as heurísticas existentes, destacam-se as bio-inspiradas, guiadas por funções que avaliam os subconjuntos de atributos gerados.

De acordo com Lee (2017), os métodos de seleção de atributos podem ser classificados em *wrapper, filter, embeddeds* ou híbrido. Os *wrappers* avaliam a qualidade de um subconjunto de atributos ao utilizá-lo no treinamento de um classificador, sendo que quanto maior a acurácia obtida (outras métricas de avaliação também podem ser empregadas), melhor ele é. Métodos *filters* utilizam medidas estatísticas para avaliar os subconjuntos e algoritmos de *clustering* para agrupá-los, sem exigir o treinamento de classificadores. Por sua vez, *Embeddeds* selecionam os atributos à medida que um algoritmo de aprendizado é treinado e construído. Logo, *wrappers* costumam ser mais lentos que os outros métodos devido ao processo de treinamento, porém são responsáveis por encontrarem subconjuntos que levam à maiores acurácias. Além disso, ao avaliar atributos em conjuntos, e não individualmente, esse método leva em consideração a influência das características entre si, segundo Bonidia (2019).

O presente trabalho busca investigar o algoritmo *filter* proposto por Bonidia (2019), onde a seleção de atributos ocorre por **rodadas** e é introduzida uma etapa de **votação**, responsável por gerar subconjuntos de características. Tendo em vista que as abordagens apresentadas são novas e o trabalho alcançou resultados

promissores, o presente trabalho buscou investigar o desempenho do algoritmo quando convertido para *wrapper,* visto que o método costuma atingir melhores resultados, e combinado com Algoritmo Genético.

MATERIAL E MÉTODOS

A heurística empregada na busca da solução ótima, ou seja, o menor conjunto de atributos relevantes, não redundantes e não ruidosos, é um algoritmo genético (AG) simples de Carvalho (1999) cujas principais características estão listadas a seguir:

a) indivíduos representados por cadeias de bits, onde '1' representa o atributo a ser mantido e '0', a ser descartado;

b) seleção por torneio;

c) crossover de um ponto;

d) critério de parada adotado: estagnação do *fitness* do melhor indivíduo da população;

e) elitismo.

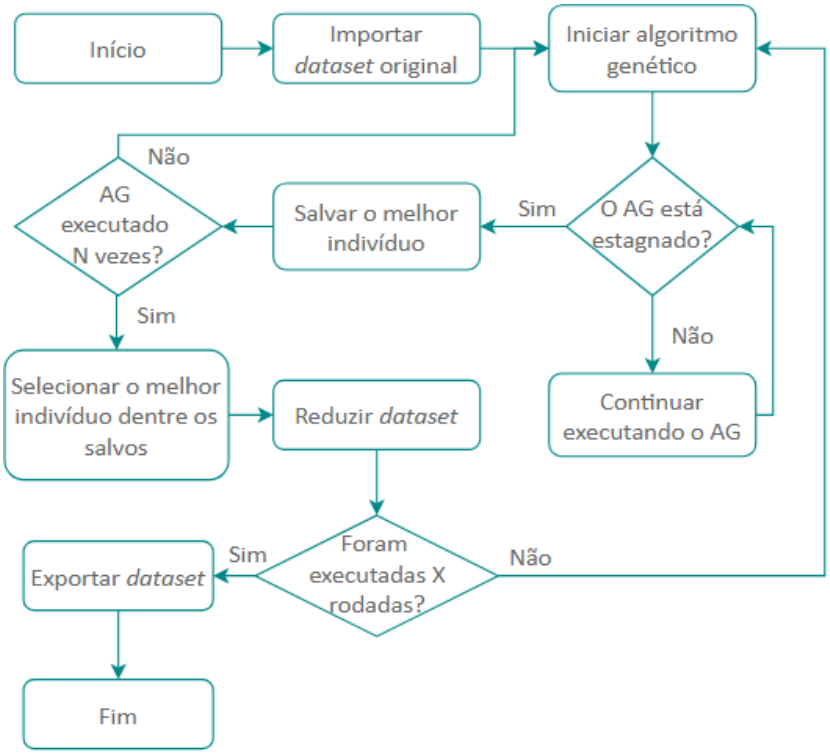
Após o algoritmo genético ser executado N vezes, são armazenados os melhores indivíduos gerados em cada execução (N melhores indivíduos). Dentre eles, seleciona-se o que possui maior *fitness* e o *dataset* é reduzido para o conjunto de atributos representados pelos seus genes. Esse processo representa uma rodada. Inicia-se então uma nova rodada, onde são retomadas as execuções do AG, porém desta vez explorando o espaço de busca do novo conjunto.

Tal algoritmo de seleção pela abordagem *wrapper* recebeu o nome de "*Best*", fazendo referência à redução do *dataset* com base no melhor (*best)* indivíduo entre os N melhores. Para facilitar a compreensão, seu ciclo de execução é ilustrado na Figura 1.

Gerou-se uma versão do *Best* acrescentando uma etapa de seleção dos atributos, a votação, que gera os chamados "indivíduos finais". O número de votos equivale à quantas vezes um atributo apareceu entre os melhores indivíduos. Os atributos que formarão um indivíduo final são aqueles que possuem uma quantidade mínima de votos.

Cada indivíduo final é formado a partir de uma quantidade mínima de votos distinta. Logo, para um vetor de quantidade mínimas de votos de tamanho K, surgem K indivíduos finais. Esse vetor é gerado da seguinte maneira: multiplica-se um vetor de porcentagens de tamanho K pela quantidade de melhores indivíduos N. Exemplo: dado o vetor de porcentagens {10%; 50% e 100%} e 20 melhores indivíduos, o vetor de votos mínimos será {2; 10 e 20}, assim, serão gerados 3 indivíduos finais, um com os atributos que aparecem no mínimo duas vezes entre os melhores indivíduos, outro com no mínimo 10 vezes e, por fim, 20 vezes (deve aparecer em todos os melhores indivíduos). Um exemplo de geração de um indivíduo final é ilustrado na Figura 2.

Figura 1 – Algoritmo *Best*



Fonte: Autoria própria (2020)

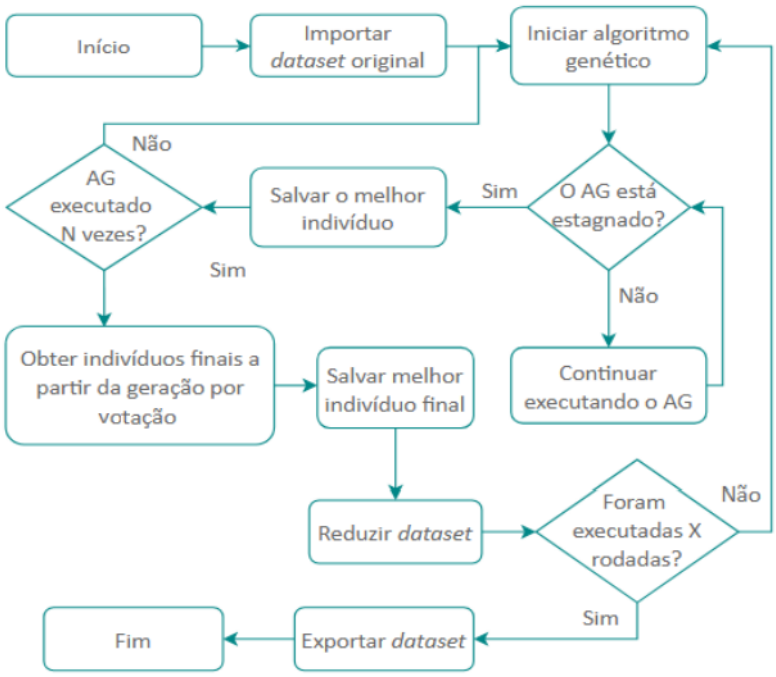
Figura 2 – Exemplo de indivíduo final gerado a partir de uma porcentagem de 50%

Melhores indivíduos	0	1	1
	1	1	1
	0	1	0
	0	1	0
	0	0	1
	1	1	0
	0	1	0
	0	0	1
Indivíduo final	**0**	**1**	**1**

Fonte: Autoria própria (2020).

Entre os K indivíduos finais, aquele que apresenta o maior *fitness* é empregado na redução do *dataset*. Por este motivo, o novo algoritmo recebe o nome de "*Final*". Sua estrutura é exibida na Figura 3.

Figura 3 – Algoritmo *Final*



Fonte: Autoria própria (2020).

Inicialmente, o valor *fitness* concedido à cada indivíduo foi equivalente à acurácia média de treinamento do classificador *Naive Bayes*, escolhido devido ao baixo tempo de treinamento e não possuir parâmetros. No entanto, também foram feitos experimentos utilizando uma função ponderada como *fitness*, com o intuito de encontrar soluções com redução ínfima da acurácia e significativa de atributos.

$$fitness = \alpha \cdot \text{acurácia} - \beta \cdot \text{nº de atributos} \quad (1)$$

Os *datasets* utilizados nos experimentos foram:

a) *Spambase* com 4601 instâncias e 56 atributos, obtido de Dheeru (2017);

b) *Waveform* (*Noise*) com 5000 instâncias e 40 atributos, obtido de Dheeru (2017);

c) *Winequality* (*White*) com 4898 instâncias e 11 atributos, obtido de Dheeru (2017);

d) *NBA (2006-2019)* com 1088 instâncias e 130 atributos, obtido de Fukano (2019).

Após executar o *Best* e o *Final*, com e sem a função ponderada, foram salvos as acurácias e os atributos dos indivíduos que mais se destacaram (entre todos os melhores indivíduos, para o caso do *Best*, e entre todos os melhores e finais indivíduos, para o *Final*).

Com a finalidade de comparar os resultados com outro método de redução, implementou-se o *Principal Component Analysis* (PCA) de Pearson (1901). Nessa etapa, o método foi executado definindo o número de componentes principais igual à quantidade de atributos. Em seguida, o conjunto de dados transformado foi submetido ao classificador iterando a quantidade de atributos, em outras palavras,

acrescentando a cada iteração 1 componente principal. Assim, foi encontrada a máxima acurácia atingida e o número de componentes necessários.

Os parâmetros estabelecidos nos testes foram:

a) quantidade máxima de gerações estagnadas: 30;

b) taxa de mutação: 0,03;

c) taxa de crossover: 0,50;

d) números de execuções do AG: 10 e 30;

e) lista de frequências mínimas: [0,1; 0,2; ...0,9; 1,0];

f) número de rodadas: 6;

g) pesos α: 100 e β: 0,1.

RESULTADOS E DISCUSSÃO

Nas tabelas, os campos que apresentam um hífen (–) indicam os *datasets* em sua forma original, enquanto que "(*fitness*)" em frente ao nome dos algoritmos indica que esses utilizaram a função ponderada.

Tabela 1 – Resultados do treinamento com os conjuntos originais e reduzidos

Dataset	Algoritmo	Atributos	Acurácia
Spambase	–	57	0,8600
	PCA	3	0,8102
	Best	16	0,9297
	Best (fitness)	13	0,9270
	Final	36	0,9227
	Final (fitness)	11	0,9119
Waveform	–	40	0,8017
	PCA	35	0,8293
	Best	17	0,8549
	Best (fitness)	12	0,8528
	Final	17	0,8549
	Final (fitness)	10	0,8488
Winequality	–	11	0,6568
	PCA	10	0,6863
	Best	4	0,6914
	Best (fitness)	4	0,6914
	Final	4	0,6914
	Final (fitness)	4	0,6914
NBA (2006 – 2019)	–	130	0,6585
	PCA	2	0,5976
	Best	16	0,7195
	Best (fitness)	5	0,7073
	Final	9	0,6829
	Final (fitness)	5	0,7317

Fonte: Autoria própria (2020).

Como mencionado anteriormente, os experimentos também foram executados aumentando a quantidade N de execuções do AG, para 3 *datasets*.

Tabela 2 – Resultados do aumento da amostragem de 10 para 30 melhores indivíduos

Dataset	Algoritmo	10 melhores indivíduos		30 melhores indivíduos	
		Atributos	Acurácia	Atributos	Acurácia
Spambase	Best	16	0,9297	17	0,9297
	Best (fitness)	13	0,9270	18	0,9269
	Final	36	0,9227	14	0,9283
	Final (fitness)	11	0,9119	24	0,9269
Waveform	Best	17	0,8549	18	0,8559
	Best (fitness)	12	0,8528	11	0,8528
	Final	17	0,8549	10	0,8478
	Final (fitness)	10	0,8488	15	0,8559
Winequality	Best	4	0,6914	4	0,6914
	Best (fitness)	4	0,6914	4	0,6914
	Final	4	0,6914	4	0,6914
	Final (fitness)	4	0,6914	4	0,6914

Fonte: Autoria própria (2020).

Ao comparar o desempenho do classificador com o *dataset* original e os reduzidos, percebe-se que em todos os casos, os algoritmos propostos alcançaram subconjuntos que melhor descrevem as classes das instâncias, pois obtiveram acurácias maiores e também menor dimensionalidade (salvo, para este último critério, o *Spambase* reduzido pelo PCA com apenas 3 componentes principais).

Analisando apenas o *Best* e o *Final*, nota-se que eles alcançam acurácias muito próximas quase que na totalidade dos experimentos, quando não equivalentes, como ocorreu com o *dataset Winequality*, o menor entre os 3 adotados, o que sugere duas hipóteses: 1) só há diferença significativa quando ambos são aplicados em conjuntos grandes de dados ou 2) não há diferença entre eles.

Quanto ao emprego da função *fitness* ponderada, é perceptível sua capacidade de guiar a busca até uma solução com acurácia um pouco menor, porém com maior redução de informações. No que diz respeito ao aumento da amostragem, as acurácias obtidas melhoraram ligeiramente em apenas alguns casos, para os *datasets Spambase* e *Waveform*, sendo que para o *Winequality* não houve diferença, por ser um conjunto pequeno.

CONCLUSÃO

O presente trabalho propôs os algoritmos de seleção de atributos *Best* e *Final,* que utilizam abordagem *wrapper* e, no caso do *Final,* rodadas e votos. Ao serem avaliados com 4 *datasets* de diferentes quantidades de características e de diferentes contextos, ambos os algoritmos se revelaram promissores, pois encontraram soluções que 1) incluíam menos atributos do que o conjunto original e 2) levaram à maiores acurácias na classificação, inclusive quando comparados com o outro método de redução PCA.

AGRADECIMENTOS

Os autores agradecem à Universidade Tecnológica Federal do Paraná – Cornélio Procópio (UTFPR-CP) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

REFERÊNCIAS

BONIDIA, R. P.; CARVALHO, A. C. P. L. F.; PASCHOAL A. R.; SANCHES, D. S. Selecting the Most Relevant Features for the Identification of Log Non-Coding RNAs in Plants. **2019 8th Brazilian Conference on Intelligent Systems (BRACIS)**, Salvador, Brazil, p. 539-544, 2019. Disponível em: https://ieeexplore.ieee.org/document/8923907. Acesso em: 20 ago. 2020.

CARVALHO, A. C. P. L. F; LACERDA, E. G. M. Introdução aos algoritmos genéticos. **Anais**. Rio de Janeiro: EntreLugar, 1999.

DHEERU, D.; CASEY, G. Machine Learning Repository (UCI). 2017. Disponível em: https://archive.ics.uci.edu/ml/index.php. Acesso em: 20 ago. 2020.

FUKANO, A. M. K. **Predição de resultados de jogos da NBA**: uma abordagem de mineração de dados com aprendizado de máquina para playoffs. 2019 Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2019.

LEE, P.; LOH, W. P.; CHIN, J. Feature selection in multimedia: the state-of-the-art review. **Image and Vision Computing**, v. 67, 2017. Disponível em: https://www.researchgate.net/publication/319961539_Feature_selection_in_multimedia_The_state-of-the-art_review. Acesso em: 19 ago. 2020.

MAFARJA, M.; MIRJALILI, S. Whale optimization approaches for wrapper feature selection. **Applied Soft Computing**, v. 62, p. 441-453, 2018. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S1568494617306695. Acesso em: 20 ago. 2020.

PEARSON, K. On lines and planes of closest fit to systems of points in space. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, v. 6, p. 559-572, 1901. Disponível em: https://www.tandfonline.com/doi/abs/10.1080/14786440109462720. Acesso em: 20 ago. 2020.