

Análise de Métodos Construtivos, Busca Local e Metaheurísticas para o Problema de Seleção de Características

Honovan Paz Rocha^{*‡}, Rafael Wilson Soares Ladeira[†] and Antônio de Pádua Braga[‡]

^{*}Instituto de Engenharia, Ciência e Tecnologia

Universidade Federal dos Vales do Jequitinhonha e Mucuri - Av. Manoel Bandejas 460, 39440-000, Janaúba, MG, Brasil

[†]Programa de Graduação em Engenharia de Controle e Automação

Universidade Federal de Minas Gerais - Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil

[‡]Programa de Pós-Graduação em Engenharia Elétrica

Universidade Federal de Minas Gerais - Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil

Resumo—As dificuldades encontradas na tarefa de classificação de padrões quando são utilizadas bases de dados com grande dimensionalidade tornam a tarefa de seleção de características essencial. Neste trabalho apresentam-se comparações de heurísticas e metaheurísticas utilizadas com o objetivo de se encontrar o menor conjunto de variáveis sem perda de eficácia na tarefa de classificação. Além da utilização de métodos clássicos para seleção, propõe-se no trabalho uma abordagem que utiliza um filtro uni-variado para rankear o conjunto total de características de forma a medir a relevância individual de cada uma em relação à discriminação entre classes, sendo este rank utilizado na etapa construtiva de um método GRASP. A utilização deste filtro caracterizou a abordagem utilizada como um método híbrido para seleção de características, devido à exigência da combinação entre um método puramente estatístico e uma avaliação baseada num classificador, para realização da estratégia de busca pelo espaço de características. Os experimentos realizados demonstraram eficácia da abordagem quanto à acurácia, mas o custo computacional é elevado quando são utilizadas bases de dados de maior dimensionalidade.

Keywords—Seleção de Características, GRASP, Busca Tabu, F-Score.

I. INTRODUÇÃO

A classificação de padrões é uma tarefa que atribui um determinado objeto (padrão) a uma categoria (classe), dado um conjunto de características (também conhecidas como variáveis ou atributos) que representam este objeto. Nesta tarefa, em resumo, determina-se a probabilidade de um objeto pertencer a uma categoria, fazendo com que geralmente seja impossível a obtenção de uma classificação ótima [1]. A tarefa de classificação tem como objetivo a obtenção de modelos com alta capacidade de generalização, isto é, com níveis aceitáveis de acurácia para dados não conhecidos pelo classificador durante o treinamento [2]. Bases de dados de problemas reais geralmente contém uma grande quantidade de dimensões, onde existem muitos atributos irrelevantes e muitas vezes um número reduzido de amostras, o que ocasiona um aumento de complexidade computacional e perda de exatidão na tarefa de classificação[3]. Nestes casos torna-se desejável a remoção de características irrelevantes e a definição de um subconjunto reduzido de características discriminativas para melhorias na classificação [4], o que conseqüentemente pode

gerar modelos para classificação com aumento da capacidade de generalização. A seleção de características é uma tarefa complexa devido à natureza combinatória do problema, sendo categorizado na classe de problemas de otimização NP-Difícil, onde um problema contendo d características possui espaço de busca 2^d [5]. Definido-se P como um conjunto de amostras com tamanho n , $F = \{f_1, f_2, \dots, f_n\}$ como sendo o conjunto total de características com cardinalidade $= d$ e $S(S \subseteq F)$ como um subconjunto de F , pode-se definir $J(S)$ como a função de avaliação que mede relevância de S em relação ao nível de discriminação entre classes com base na acurácia. Pode-se definir o problema de seleção de características como sendo a busca por um conjunto X de maneira que:

$$J(X) = \max_{S \subseteq F} J(S) \quad (1)$$

Neste trabalho utiliza-se outra maneira de definir o problema de seleção de características em termos do modelo de otimização como descrito pela Equação 2. Nesta formulação o objetivo é encontrar o menor subconjunto de características com acurácia ($C(S)$) maior que uma solução referência (ϵ), que é obtida a partir da aplicação de um classificador ao conjunto de dados utilizando todas as características.

$$\begin{aligned} \min J(X) &= |S| \\ \text{sujeito a:} \\ S &\subseteq F, C(S) > \epsilon \end{aligned} \quad (2)$$

Por ser um problema NP-Difícil a única maneira de se garantir a seleção de uma solução ótima é através de uma busca exaustiva pelo espaço de soluções, o que dependendo da quantidade de variáveis é impraticável. A utilização de heurísticas e metaheurísticas são alternativas para se encontrar boas soluções sem a necessidade de exploração de todo o espaço de busca. Em aplicações do mundo real estas técnicas têm se mostrado adequadas às necessidades visto que uma solução é satisfatória quando tem qualidade aproximada à uma ótima mas encontrada com tempo factível [6]. Neste contexto, propõe-se neste trabalho a comparação de heurísticas e metaheurísticas com diferentes estratégias de busca com o objetivo de selecionar as K melhores características dentro do conjunto total e que, não pioram a classificação quanto à taxa

de acertos, tendo como referência a solução contendo todas as características. São utilizados métodos construtivos clássicos: SFS (em inglês, *sequential forward selection*), SBE (em inglês, *Sequential Backward Elimination*) SFFS (em inglês, *Sequential Forward Floating Selection*) [2], métodos baseados em busca local em amplitude e profundidade com diferentes estruturas de vizinhança e as metaheurísticas Busca Tabu [7] e GRASP (em inglês, *Greedy Randomized Adaptive Search Procedure*) [8], onde esta última é modificada de maneira a implementar um filtro univariado na etapa de construção da solução, sendo então considerado um método híbrido de seleção por combinar uma abordagem *wrapper* utilizando o classificador de Bayes [1] para avaliação de soluções e o filtro *F-Score* [9] que também avalia soluções mas sem a necessidade de um classificador. O trabalho está organizado da seguinte maneira: a seção 2 detalha algumas técnicas para seleção de características, a seção 3 apresenta trabalhos relacionados, a seção 4 apresenta o metodologia proposta, na seção 5 são descritos os experimentos computacionais realizados e, por fim, a seção 6 expressa as conclusões do trabalho.

II. TÉCNICAS PARA SELEÇÃO DE CARACTERÍSTICAS

No contexto de classificação as técnicas de seleção de características diferem quanto à forma utilizada para incorporar a busca no espaço adicional dos subconjuntos de características, à escolha do modelo, dividindo-se em três categorias: métodos de filtro, métodos *wrapper* e métodos embarcados [10]. Os métodos de filtro e *wrapper* se diferem na forma de avaliação dos subconjuntos de características. Os filtros utilizam critérios baseados em informações intrínsecas aos dados sem utilização de nenhuma técnica de aprendizagem de máquina enquanto que *wrappers* utilizam o desempenho de uma máquina de aprendizagem treinada utilizando um subconjunto específico de características. Os métodos de filtros são também conhecidos como métodos de ranqueamento de características, pois na maioria dos casos realizam o cálculo de um índice de relevância das características em relação à discriminação obtida em relação às categorias encontradas nos dados. Filtros e *wrappers* também podem ser combinados para formarem técnicas híbridas onde se utilizam os filtros para criação do rank e, em seguida, utiliza-se uma abordagem *wrapper* levando em consideração as características mais relevantes. Estas duas técnicas necessitam de uma estratégia de busca para explorar o espaço de subconjuntos de características devido à inviabilidade de se efetuar uma busca exaustiva num espaço com muitas dimensões. Nos métodos embarcados a busca por um subconjunto ótimo de características é realizada dentro do processo de construção da máquina de aprendizagem [2]. Como detalhado em [6] uma abordagem *wrapper* pode ser utilizada com estratégias de busca construtivas visando encontrar boas soluções e com custo computacional factível. Os métodos SFS e SBE são métodos clássicos bem conhecidos e de simples implementação com uma tendência conhecida a ficar preso em mínimos locais mas mesmo assim são muito utilizados em abordagens *wrapper* pelo baixo custo computacional quando comparados a métodos estocásticos avançados de seleção [11], além disso, são métodos geralmente encontrados em softwares estatísticos conhecidos como o SPSS e o BMDP [12]. No SFS inicia-se com um subconjunto vazio e adiciona-se características de forma sequencial com base na melhoria gerada a partir da acurácia obtida por um classificador até que

se atinja a quantidade desejada de características. O SBE se caracteriza pela operação inversa onde o subconjunto inicial contém todas as características, sendo que a menos promissora em relação ao desempenho do classificador é eliminada de maneira progressiva até que se encontre a quantidade de variáveis desejada. Estes métodos sofrem do problema denominado efeito de aninhamento, em que no caso do SFS uma característica adicionada não pode mais ser eliminada e no SBE uma característica eliminada não pode ser selecionada novamente, fazendo com que estes métodos sejam apenas sub-ótimos. Uma maneira de contornar essas dificuldades foi proposta pelo método de seleção sequencial adiante flutuante (SFFS) [13] que consiste em alternar entre o funcionamento do SFS, adicionando características aprimorantes e o funcionamento do SBE, eliminando a característica menos promissora progressivamente. O SFFS pode ser encontrado em diversos trabalhos [6], [2] sendo considerado em [14] como sendo uma das técnicas mais efetivas para seleção de características.

Neste trabalho utilizou-se um método híbrido para seleção de características combinando-se as informações de um filtro e uma abordagem *wrapper*. O filtro utilizado é baseado em análise uni-variada, onde se realiza uma análise relativa à relevância individual de cada uma das características considerando-se independência entre elas. O filtro utilizado e classificador acoplado ao método *wrapper* são descritos a seguir.

A. F-Score

O filtro *F-Score* (*Fisher score*) é um critério simples e eficiente que, através de características estatísticas dos dados, mede a relevância das características para discriminação entre classes [9]. Considerando-se um problema de classificação binário com as classes C_1 e C_2 , ele é definido por:

$$f(i) = \frac{(\mu_i^{c_1} - \mu_i) + (\mu_i^{c_2} - \mu_i)}{\sigma_i^{c_1} + \sigma_i^{c_2}} \quad (3)$$

onde i corresponde ao índice da i -ésima características e, μ_i^c e σ_i^c são média e desvio padrão para a classe C em relação à característica i .

B. Classificador de Bayes

O Classificador de Bayes baseia-se na suposição de que o problema de decisão é visto de uma forma probabilística onde se conhecem todos os valores de probabilidades relevantes. A classificação de um objeto a uma determinada classe é feita de acordo com a probabilidade de o objeto pertencer à classe [1]. Um classificador de Bayes simples (também denominado classificador de Bayes ingênuo) supõe independência entre as variáveis, o que não ocorre na maioria dos problemas de classificação, mas ainda assim obtém resultados competitivos com a maioria dos classificadores além de possuir menor complexidade computacional devido à facilitação nos cálculos utilizados obtida pela suposição de independência. A fórmula geral utilizada pelo classificador de Bayes é dada por:

$$P(C_j|X) = \frac{P(X|C_j)P(C_j)}{P(X)} \quad (4)$$

onde $P(C_j|X)$ é o termo definido como probabilidade à posteriori que indica a probabilidade da classe ser C_j dado que o padrão X foi mensurado. O termo $p(X|C_j)$ é uma

probabilidade condicional denominada verossimilhança que representa a probabilidade de X dado que a classe C_j foi apresentada e $P(C_j)$ é a probabilidade a priori, sendo a informação que reflete o conhecimento prévio que se tem sobre os dados em relação à predição de determinado objeto pertencer à classe C_j levando em consideração apenas as quantidades de objetos amostrados em cada classe. O termo $p(X)$ é definido como evidência e pode ser visto como um mero fator de escala que garante que a soma das probabilidades à posteriori é igual a um.

III. TRABALHOS RELACIONADOS

Alguns trabalhos na literatura utilizam heurísticas para seleção de características. Alguns destes, com características semelhantes às presentes neste trabalho, serão brevemente descritos.

Em [15] foi proposto um método para selecionar e atribuir pesos às características utilizando uma abordagem *wrapper*, onde a busca tabu é utilizada como estratégia de busca e a performance do classificador knn para avaliação de subconjuntos candidatos. Os pesos encontrados são utilizados durante a classificação, fazendo com que a busca ocorra de forma mais intensa em dimensões relacionadas a características com maior peso. As características foram codificadas de forma binária e uma solução do problema é composta por três partes: pesos, características e um valor k , que corresponde aos k vizinhos utilizados no knn. A função objetivo é definida em termos da acurácia do classificador e a vizinhança é definida pela modificação de um peso ou característica, sendo k atribuído de forma aleatória a cada vizinho.

No trabalho de Yusta [6] foram utilizados os métodos GRASP, busca tabu e algoritmos meméticos para seleção de características. Estes métodos foram comparados a métodos clássicos para este problema, incluindo-se algoritmos evolutivos. Essegir [16] utilizou como base o algoritmo descrito em [6], promovendo modificações que diminuíram o tempo computacional através da utilização dos filtros relief [17], incerteza Simétrica (IS) [2] e FCBF [18] na fase de construção do método GRASP e definição de outras funções para geração da vizinhança na etapa de busca local.

IV. METODOLOGIA

Os métodos implementados neste trabalho consistem em três métodos construtivos (SFS, SBE e SFFS), duas buscas locais do tipo descida mais rápida com funções vizinhança diferentes e 2 metaheurísticas, Busca Tabu e GRASP-FS (em inglês, *GRASP for feature selection*). As buscas locais e a Busca Tabu utilizam como solução inicial aquelas encontradas pelo método SFS. Todos os métodos realizam uma busca pelo maior valor de acurácia para uma quantidade fixa de características. Esta quantidade variou da quantidade total até um conjunto contendo apenas uma característica. A seguir são descritos os algoritmos de busca e metaheurísticas utilizadas com as respectivas modificações efetuadas.

A. Buscas Locais

Uma busca local consiste numa família de métodos que realiza uma busca pelo espaço de soluções com base no conceito de vizinhança [19]. Esta busca é realizada de forma

iterativa até que se encontre um mínimo local. Sendo F o espaço total de busca, determina-se uma função N que associa uma solução $s \in F$ à sua vizinhança $N(s) \subseteq F$. Desta maneira pode-se descrever uma solução s' como sendo vizinha de uma solução s se esta solução foi obtida a partir de uma modificação m definida pela operação $s' \leftarrow s \oplus m$. Assim, cada solução $s' \in N(s)$ é um vizinho de s . Neste trabalho foram implementadas duas buscas locais para seleção de características em que estas se diferenciam pela função de vizinhança. Utilizou-se o método da descida íngreme (em inglês, *Steepest Descent - SD*) como heurística de busca, que consiste em analisar todos os vizinhos de uma solução s e selecionar aquele melhor aprimorante em relação à função de avaliação. O método não possui critério de parada sendo que a busca é finalizada quando não se encontra nenhum vizinho aprimorante na vizinhança, consistindo assim numa solução de mínimo local. O primeiro método de busca local implementado tem como base a estrutura de vizinhança utilizada em [6] definida pela Equação 5. Esta estrutura de vizinhança, denominada *Attribute Flip - AF* em [16] consiste em uma permutação entre uma característica presente na solução atual e outra ainda não selecionada. Considerando-se a representação das variáveis como 1 para uma característica presente na solução e 0 para uma solução ausente esta função de vizinhança opera sobre $k * (|F| - k)$ diferentes soluções tornando-se impraticável em problemas com alta cardinalidade.

$$NH(S) = \{X, X = S \cup \{f_i\} \setminus \{f_j\}, \forall f_i \in F \setminus S, \forall f_j \in S\} \quad (5)$$

A outra estrutura de vizinhança utilizada tem como base a função SFFS1 que demonstrou bons resultados em [16], onde foi descrita. A SFFS1 toma como base as operações realizadas pelo método construtivo SFFS. A vizinhança é definida pela adição de uma característica que aprimore a solução atual e, em seguida, elimina a característica que gera o menor benefício. Esta função de vizinhança opera sobre $((|F| - k) + (|F| - k))$ soluções, tendo menor custo que a estrutura AF ao analisar toda a vizinhança de uma solução.

B. Busca Tabu

O método Busca Tabu descrito em [20] e [21] deriva-se da busca local descida mais íngreme, mas não termina a busca quando se encontra um mínimo local, pois aceita também soluções não aprimorantes como o objetivo de escapar de mínimos locais. De maneira geral a busca ocorre em torno da seleção do melhor vizinho $s' \in N(s)$ de uma solução s , e toma este como nova solução mesmo $J(s')$ não sendo melhor que $J(s)$. Com esta estratégia pode-se escapar de mínimos locais mas para evitar voltar à solução anterior numa próxima iteração utiliza-se o conceito de lista Tabu, evitando assim a geração de ciclos. Elementos adicionados a esta lista tem movimentos m relacionados a eles proibidos por um certo número de iterações, é o chamado tempo tabu. Alguns movimentos tabu são permitidos segundo um critério de aspiração como por exemplo, gerar uma solução melhor que a melhor solução obtida até o instante atual da busca. A busca ocorre até que um critério de parada, uma quantidade de iterações sem melhora por exemplo, seja satisfeito. Os parâmetros tempo tabu e critério de parada utilizados neste trabalho são baseados em [6].

C. GRASP (Greedy Randomized Adaptive Search Procedure)

O GRASP [8] é um método baseado em múltiplas inicializações que consiste em duas fases bem definidas: uma fase em que se constrói a solução adicionando-se elemento a elemento de maneira iterativa, e outra onde se refina, através de busca local, a solução gerada na etapa de construção. Este processo de construção e busca ocorre por p iterações onde a melhor solução é retornada ao final do processo.

A fase de construção do método consiste na geração de uma solução viável em que cada elemento candidato é adicionado de forma progressiva até que a solução esteja completa. Os elementos candidatos a compor a solução são ordenados com base numa função gulosa que mede o benefício gerado pela adição do elemento à solução parcial. Defini-se então um subconjunto dos melhores candidatos para comporem uma lista denominada lista restrita de candidatos (LRC) cujo tamanho é determinado por um parâmetro $\alpha \in [0, 1]$, onde para $\alpha = 0$ obtém-se um comportamento puramente guloso do algoritmo enquanto que com $\alpha = 1$ tem-se um comportamento totalmente aleatório. Uma solução é obtida através da seleção aleatória de um elemento presente em LRC, fazendo com que se possa gerar diferentes soluções viáveis a cada iteração do método. A função gulosa utilizada neste trabalho baseia-se no índice de relevância gerado para cada elemento a partir do filtro *F-Score*. O algoritmo Busca Tabu é utilizado na etapa de busca local com estrutura de vizinhança AF. Os demais parâmetros do algoritmo foram utilizados como em [6]. A fase de construção do GRASP tem complexidade na ordem de $O(d^2 * n)$ enquanto que a fase de busca local tem $O(d^2 * maxIter * n)$ o que consequentemente gera uma complexidade para o método GRASP de $graspMax * (O(d^2 * n) + O(d^2 * maxIter * n))$. As variáveis *maxIter* e *graspMax* estão definidas em forma de comentário no pseudo-código a seguir, que mostra uma visão geral da estrutura utilizada considerando-se a versão do método GRASP com o filtro F-score.

Algorithm 1 Pseudo-código do Método - GRASP-KFeatures

```

( $X_t, D_t$ )  $\leftarrow$  Carregar Conjunto de Treinamento;
( $X_v, D_v$ )  $\leftarrow$  Carregar Conjunto de Validação;
 $alphaGrasp$   $\leftarrow$  Parâmetro alfa do GRASP;
 $graspMax$   $\leftarrow$  No. Soluções Avaliadas GRASP;
 $btMax$   $\leftarrow$  No. Iterações sem melhoria - Busca Tabu;
 $maxIter$   $\leftarrow$  No. Max Iterações Busca Tabu;
 $tabuTime$   $\leftarrow$  Tamanho Lista Tabu;

```

```

 $accR$   $\leftarrow$  ClassificadorBayes( $X_t, D_t, X_v, D_v$ );
 $rankFeatures$   $\leftarrow$  fScore( $X_t, D_t$ );

```

```

 $k$   $\leftarrow$   $|S| - 1$ ;
while ( $acc > accR$ ) do
    ( $subset, acc$ )  $\leftarrow$  GRASP ( $X_v, D_v, rankFeatures,$ 
         $alphaGrasp, graspMax, btMax, maxIter,$ 
         $tabuTime$ );
     $k$   $\leftarrow$   $k - 1$ ;
end while

```

```

Calcular Acurácia em relação ao  $subset$ ;

```

V. EXPERIMENTOS COMPUTACIONAIS

As heurísticas e o classificador utilizados na metodologia proposta foram implementados em Matlab versão 7.10. Os experimentos computacionais foram realizados em uma máquina Intel Core2Duo 2.1Ghz, com 4 GB de Ram e Sistema Operacional Windows 7. As bases de dados utilizadas para classificação são todas binárias e foram obtidas em [22]. A primeira é a base ionosfera, que contém 351 amostras e 34 atributos. A segunda base é a do câncer de mama de Wisconsin, que consiste de 569 amostras e 31 atributos. A terceira base é a de análise de crédito da Alemanha que tem 1000 amostras e 24 atributos. A quarta base é a de doenças do coração, que é formada por 270 padrões e 13 atributos. As bases de dados foram divididas de forma que utilizou-se 50% dos dados para treinamento, 25% para validação e 25% para testes, sendo que estas amostragens foram realizadas de maneira aleatória e mantendo-se a proporção entre as classes.

Para o treinamento do classificador de Bayes utilizou-se a porção dos dados definida como conjunto de treinamento, com 50% dos dados, enquanto que, os conjuntos validação e teste continham 25% dos dados cada um, sendo que o subconjunto de validação é utilizado para definir a exatidão do classificador de Bayes no método *wrapper* e o subconjunto de testes é utilizado num momento posterior, após a seleção final de características, com o objetivo de testar o desempenho do método proposto. Ao fazer o treinamento obteve-se os valores de acurácia 63.64%, 93.66%, 71.20% e 85.29% para as bases ionosfera, câncer, crédito e coração respectivamente.

A Tabela I mostra os indicadores valor de acurácia referencial (AccRef), quantidade de características para o maior valor de acurácia (K), maior valor de acurácia (MaxAcc), ganho percentual em relação à solução referencial (Gain) e tempo de cpu (CpuTime) para cada método testado em cada uma das bases de dados. O valor MaxAcc para o método GRASP-FS consiste no maior encontrado para 10 execuções do método e quantidade de característica é o valor da moda. As informações relativas à média e desvio padrão para o GRASP-FS não são mostradas devido ao método ter alcançado o mesmo valor de acurácia em todas as execuções, demonstrando estabilidade neste quesito.

A partir dos experimentos realizados pode-se visualizar que para a base de dados Ionosfera o valor da máxima acurácia foi o mesmo para 6 dos 7 métodos, sendo que apenas o SBE não encontrou o valor, mas o método GRASP-FS encontrou o maior valor com a menor quantidade de características, K=7. Em relação à base do Câncer o resultado foi semelhante ao da base Ionosfera, onde apenas o método SBE não ficou junto aos outros no quesito máxima acurácia, sendo que considerando-se 6 melhores, a Busca Tabu junto ao GRASP-FS obtiveram a menor quantidade de características, com K=4. Na base de análise de crédito o melhor resultado obtido quanto à acurácia foi do GRASP-FS mesmo não conseguindo o menor valor de K. Na base do coração pode-se observar um resultado um pouco menos esperado, onde junto ao GRASP-FS e à Busca Tabu, está o método SBE com melhor resultado de MaxAcc e com apenas uma característica a mais que a quantidade encontrada pelos outros dois métodos.

É possível observar pelos experimentos que o método GRASP-FS não foi superado por nenhum dos outros métodos

Tabela I. ACURÁCIA OBTIDA PARA AS BASES DE CLASSIFICAÇÃO UTILIZANDO CADA MÉTODO

		Ionosfera	Câncer	Crédito	Coração
SFS	AccRef	63, 64	93, 66	71, 20	85, 29
	K	12	7	10	5
	MaxAcc	97,73	98,59	78, 80	85, 29
	Gain (%)	53, 57	5, 27	10, 67	0, 00
	CpuTime(s)	38, 45	43, 15	29, 87	1, 88
SBE	AccRef	63, 64	93, 66	71, 20	85, 29
	K	9	3	13	10
	MaxAcc	95, 45	97, 89	77, 20	91,18
	Gain (%)	49, 99	4, 51	8, 43	6, 90
	CpuTime(s)	67, 76	71, 84	48, 01	2, 31
SFFS	AccRef	63, 64	93, 66	71, 20	85, 29
	K	9	6	10	4
	MaxAcc	97,73	98,59	78, 80	85, 29
	Gain (%)	53, 58	5, 26	10, 67	0, 00
	CpuTime(s)	106, 87	111, 34	29, 87	4, 87
LS-AF	AccRef	63, 64	93, 66	71, 20	85, 29
	K	11	6	10	12
	MaxAcc	97,73	98,59	78, 80	89, 71
	Gain (%)	53, 57	5, 27	10, 67	5, 17
	CpuTime(s)	73, 19	70, 23	61, 11	3, 69
LS-SFFS1	AccRef	63, 64	93, 66	71, 20	85, 29
	K	11	6	9	12
	MaxAcc	97,73	98,59	78, 80	89, 71
	Gain (%)	53, 57	5, 27	10, 67	5, 17
	CpuTime(s)	40, 63	44, 36	35, 60	3, 02
Busca Tabu	AccRef	63, 64	93, 66	71, 20	85, 29
	K	11	4	14	9
	MaxAcc	97,73	98,59	80	91,18
	Gain (%)	53, 57	5, 27	12, 36	6, 90
	CpuTime(s)	170, 88	177, 26	162, 53	6, 50
GRASP-FS	AccRef	63, 64	93, 66	71, 20	85, 29
	K	7	4	12	9
	MaxAcc	97,73	98,59	80,40	91,18
	Gain (%)	53, 57	5, 27	12, 92	6, 90
	CpuTime(s)	734, 95	1134, 95	537, 60	45, 56

em nenhuma das bases de dados utilizadas. Os resultados deste método foram bons tanto no quesito acurácia quanto em relação à quantidade de características selecionadas. Com estes resultados pode-se comprovar a eficiência do método híbrido que utiliza um método de filtro na etapa de construção e a Busca Tabu na parte de busca local. A componente aleatória que gera a necessidade de se executar os passos do método por várias iterações faz com que o tempo computacional seja muito maior em relação aos demais comparados, mas este pode ser considerado um fator de menor impacto (exceto quando for impraticável), dado que a tarefa de seleção de características é um tipo de pré-processamento para a tarefa de classificação, onde esta sim depende muito do tempo. As buscas locais utilizadas demonstraram eficiência ao encontrar bons valores de acurácia em pelo menos duas bases de dados, sendo importante observar que elas obtiveram quase os mesmos resultados tanto para MaxAcc quanto para K mas a busca LS-SFFS1 efetua um número menor de operações para encontrar tais resultados. Em relação aos métodos construtivos clássicos os resultados foram razoáveis, exceto para o SBE que obteve um bom resultado apenas para a base de dados do coração, provavelmente por ser a menor base de dados e com espaço de busca factível às limitações do SBE.

Outra informação importante surge em relação aos ganhos gerados em relação à utilização de todo o conjunto de características (AccRef). Pode-se verificar que todos os métodos obtiveram bons percentuais de ganho para quase todas as bases de dados, exceto para a base do coração onde o SFS e o SFFS não geraram ganhos em relação à acurácia, mas mesmo assim obtiveram o mesmo desempenho selecionando um conjunto

reduzido de características.

VI. CONCLUSÕES

Os resultados dos experimentos mostraram que as heurísticas utilizadas obtiveram bons resultados quanto aos modelos obtidos ao encontrar conjuntos reduzidos de características com boas taxas de acurácia. O GRASP-FS encontrou os melhores resultados e mostrou-se um método muito promissor para seleção em bases de dados com dimensionalidade moderada, sendo que este pode se tornar impraticável com grandes bases de dados. A busca local LS-SFFS1 mostrou-se também promissora ao obter bons resultados e utilizando uma estrutura de vizinhança com menor custo que a LS-AF, sendo esta última muito comum na literatura como detalhado em [6] e [16]. Os métodos clássicos demonstram resultados razoáveis mas podem ter sido responsáveis pelo bom rendimento das buscas locais que utilizaram uma solução inicial gerada por eles. Propõe-se como trabalhos futuros a utilização de outras métricas como função avaliação, que levem em consideração a capacidade de generalização e não somente a acurácia para os dados conhecidos. Seria interessante também a comparação dos métodos utilizados com algoritmos evolutivos, sendo que estes são muito utilizados em problemas de seleção. A utilização de bases de dados maiores também poderia trazer novas informações em relação aos limites dos métodos.

AGRADECIMENTOS

O presente trabalho foi realizado com o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq - Brasil. Processo n: 158265/2014-9.

REFERÊNCIAS

- [1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley and Sons, 2001.
- [2] I. Guyon and A. Elisseeff, *An Introduction to Feature Extraction*. Springer, 2006.
- [3] H. Liu and H. Y. Motoda, "Feature selection for knowledge discovery and data mining," *Kluwer Academic, Boston*, 1998.
- [4] F. Tan, X. Fu, Y. Zhang, and A. Bourgeois, "Improving feature subset selection using a genetic algorithm for microarray gene expression data," *IEEE Congress on Evolutionary Computation*, pp. 16–21, 2006.
- [5] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence* 97, pp. 273 – 324, 1997.
- [6] S. C. Yusta, "Different metaheuristic strategies to solve the feature selection problem," *Pattern Recognition Letters* 30, pp. 525 – 534, 2009.
- [7] F. Glover and M. Laguna, "Tabu search," *Kluwer Academic Publishers, Boston*, 1997.
- [8] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Global Optimization* 2, pp. 1 – 27, 1995.
- [9] Y. Chang and C. Lin, "Feature ranking using linear svm," *WCCI2008 Workshop and Conference Proceedings*, vol. 3, pp. 53–64, 2008.
- [10] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics* 23, pp. 2507 – 2517, 2007.
- [11] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, pp. 1157 – 1182, 2003.
- [12] J. Pacheco, S. Casado, and L. Núñez, "A variable selection method based on tabu search for logistic regression models," *European Journal of Operational Research* 199, pp. 506 – 511, 2009.
- [13] P. Pudil, J. Novovicová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters* 15, pp. 1119 – 1125, 1994.

- [14] A. K. Jain and D. Zongker, "Feature selection: Evaluation, application and small sample performance," *IEEE Transactions. Pattern Anal. Mach. Intell.* 19, pp. 153 – 158, 1997.
- [15] M. A. Tahir, A. Bouridane, and F. Kurugollu, "Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier," *Pattern Recognition Letters*, pp. 438 – 446, 2006.
- [16] M. A. Esseghir, "Effective wrapper-filter hybridization through grasp schemata," *JMLR: Workshop and Conference Proceedings*, pp. 45 – 54, 2010.
- [17] M. Robnik-Sikonja and I. Kononenko, "An adaptation of relief for attribute estimation in regression," *ICML: International Conference on Machine Learning*, pp. 296 – 304, 1997.
- [18] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," *ICML: International Conference on Machine Learning*, pp. 856 – 863, 2003.
- [19] H. Hoos and T. Stutzle, *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann Publishers Inc, 2004.
- [20] F. Glover, "Tabu search: Part i," *ORSA Journal on Computing*. 1, pp. 190 – 206, 1989.
- [21] —, "Tabu search: Part ii," *ORSA Journal on Computing*. 2, pp. 4 – 32, 1990.
- [22] C. L. Blake and C. J. Merz. (1998) Uci repository of machine learning databases. Irvine,CA: University of California, Dept. of Information and Computer Science. [Online]. Available: <http://www.ics.uci.edu/mlearn/MLRepository.html>