Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 27, Num. 2 (2020) 30-41

RESEARCH ARTICLE

Unsupervised Feature Selection Methodology for Clustering in High Dimensionality Datasets

Metodologia Não-Supervisionada de Seleção de Atributos para Clustering em Conjuntos de Dados de Alta Dimensionalidade

Marcos de Souza Oliveira[1*], Sérgio Ricardo de Melo Queiroz[1]

Resumo: Feature selection is an important research area that seeks to eliminate unwanted features from datasets. Many feature selection methods are suggested in the literature, but the evaluation of the best set of features is usually performed using supervised metrics, where labels are required. In this work we propose a methodology that tries to aid data specialists to answer simple but important questions, such as: (1) do current feature selection methods give similar results? (2) is there is a consistently better method ? (3) how to select the m-best features? (4) as the methods are not parameter-free, how to choose the best parameters in the unsupervised scenario? and (5) given different options of selection, could we get better results if we fusion the results of the methods? If yes, how can we combine the results? We analyze these issues and propose a methodology that, based on some unsupervised methods, will make feature selection using strategies that turn the execution of the process fully automatic and unsupervised, in high-dimensional datasets. After, we evaluate the obtained results, when we see that they are better than those obtained by using the selection methods at standard configurations. In the end, we also list some further improvements that can be made in future works.
Keywords: Feature Selection — Clustering — Dimensionality Reduction — Unsupervised Learning

Resumo: A seleção de atributos é uma importante área de pesquisa que busca eliminar variáveis indesejadas de conjuntos de dados. Muitos métodos de seleção de atributos são propostos na literatura, porém a avaliação do melhor conjunto de atributos é frequentemente realizada através de critérios supervisionados, onde as classes são exigidas. Neste trabalho é proposta uma metodologia que tentará ajudar especialistas de dados a responder questões simples, mas importantes, como: (1) os métodos existentes possuem um resultado similar? (2) existe um método consistentemente "melhor"? (3) como selecionar os m-melhores atributos? (4) dado que os métodos não são livres de parâmetros, como selecionar bons parâmetros em um cenário não-supervisionado? e (5) tendo diferentes opções de seleção, poderíamos obter melhores resultados se combinarmos os resultados dos métodos? Se sim, como podemos então combinar esses resultados? Neste trabalho, analisaremos essas questões e propomos uma metodologia que, com base em alguns métodos não-supervisionados, realizará a seleção dos atributos, utilizando estratégias que tornam a execução do processo totalmente automática e não-supervisionada, em conjuntos de dados de alta dimensionalidade. Após avaliarmos os resultados obtidos, foi possível detectar uma melhor performance em relação à obtida pelos métodos em suas configurações padrão. Ao final também elencamos algumas melhorias que podem ser realizadas em trabalhos futuros.
Palavras-Chave: Seleção de Atributos — Clustering — Redução de Dimensionalidade — Aprendizagem Não-Supervisionada

[1] *Centro de Informática, Universidade Federal de Pernambuco, Brasil*
*__Corresponding author__: marcosd3souza@gmail.com
DOI: http://dx.doi.org/10.22456/2175-2745.96081 • **Received:** 01/09/2019 • **Accepted:** 12/03/2020

1. Introdução

Métodos de redução de dimensionalidade têm se tornado uma importante área de pesquisa nos últimos anos. Seu desafio consiste na eliminação de variáveis consideradas irrelevantes/redundantes em conjuntos de dados de alta dimensionalidade. Em algumas áreas como bioinformática, processamento de imagens e classificação de textos, há situações em que uma quantidade relativamente pequena de instâncias são descritas por um número muito elevado de atributos. Isso nos coloca numa das piores situações daquilo conhecido como *The Curse of Dimensionality* [1].

A aplicação direta de algoritmos supervisionados ou não-supervisionados (e.g. *Clustering*), sem a realização de qualquer pré-processamento a fim de eliminar atributos irrelevantes, poderá resultar em problemas de performance ou na geração de resultados aleatórios (i.e. que não refletem qualquer padrão real nos dados), além de dificultar a compreensão do modelo gerado. Desta forma, surge a necessidade da aplicação de métodos para extração/seleção de atributos mais informativos.

Quando falamos em extração de atributos, consideramos a tarefa de realizar uma espécie de transformação no conjunto de dados original, um exemplo de algoritmo é o de análise de componentes principais (PCA). Já a seleção de atributos consiste em utilizar o conjunto original de variáveis, guardando aquelas mais relevantes, geralmente selecionadas a partir de um critério estatístico. O uso da seleção ao invés da extração de atributos pode facilitar a interpretação dos resultados em relação às variáveis originais dos dados. Neste trabalho trataremos da seleção de atributos.

Métodos de seleção de atributos são divididos em supervisionados e não-supervisionados. Os métodos supervisionados avaliam os atributos utilizando as classes dos objetos como uma função objetivo, otimizando critérios como o ganho de informação [2]. Os não-supervisionados utilizam alguma propriedade estatística, sem necessidade de informação a respeito da classe dos objetos.

Na literatura são propostos vários métodos para a seleção de atributos de forma não-supervisionada. Tais métodos geram uma espécie ranking para os atributos, atribuindo uma pontuação para cada atributo. Desta maneira, a forma mais comum de seleção usando esses métodos é manter os m-melhores atributos. Surge assim a necessidade de estabelecer uma estratégia para a escolha do valor de m, de forma a se obter um bom ponto de corte. Este é um problema pouco tratado pelos métodos, pois frequentemente os trabalhos selecionam as variáveis a partir de valores pré-determinados para o m, tais como $\{10, 15, 20, 25 \ldots\}$, guiando-se por um critério supervisionado (ou seja, que necessita da informação das classes dos objetos), como a precisão [2].

Além da escolha do ponto de corte, outro problema observado é que os métodos não são totalmente livres de parâmetros, surgindo a necessidade da aplicação de alguma abordagem para a escolha de uma "boa" configuração. Por fim, existe também o problema de qual método utilizar, uma vez que temos a disponibilidade de uma variedade de métodos. Várias questões portanto se impõem. Elencamos então cinco questões consideradas principais neste trabalho:

(Q1) Há métodos de seleção com resultados similares, de tal forma que possam ser considerados equivalentes?

(Q2) Há algum método consistentemente "melhor" do que os outros para diferentes conjuntos de dados com alta dimensionalidade, de forma que poderíamos considerá-lo o método "padrão" de seleção?

(Q3) Como selecionar, de forma não-supervisionada, um bom ponto de corte m para ser utilizado na escolha das m-melhores variáveis?

(Q4) Como selecionar, de forma não-supervisionada, bons parâmetros para métodos, tendo em vista que os métodos utilizados não são totalmente livres de parâmetros?

(Q5) É possível obter resultados melhores se os resultados de diferentes métodos forem combinados para obter um novo ranking de atributos? Como podemos realizar essa combinação de forma eficiente?

Diante desses questionamentos, propomos, neste trabalho, uma metodologia para a seleção de atributos em um cenário não-supervisionado, onde utilizaremos um algoritmo clássico do tipo *k-means* [3], para a atividade de *Clustering* em conjuntos de dados de alta dimensionalidade. Este trabalho será organizado da seguinte forma: a próxima seção irá apresentar alguns métodos de seleção não-supervisionados da literatura e utilizados neste trabalho, a seção 3 irá descrever a nossa metodologia, a seção 4 irá mostrar os resultados obtidos a partir dos experimentos realizados e a seção 5 irá relatar as conclusões e alguns trabalhos futuros.

2. Revisão da Literatura

Existem três categorias de métodos para a seleção de atributos: *filter*, *wrapper* e *embedded* [4]. Métodos do tipo *filter* realizam a seleção sem a necessidade de treinamento de algoritmos de aprendizagem, é utilizado alguma propriedade estatística própria dos atributos, tais como a seleção a partir da variância (os atributos menos variantes são eliminados do conjunto) uma matriz de Gauss e Laplace [5], esses métodos tendem a selecionar atributos redundantes por não avaliar correlações nos subconjuntos gerados. Métodos *filter* também tendem a ser mais rápidos.

Métodos do tipo *wrapper* utilizam algum tipo de algoritmo de aprendizagem para escolher as variáveis, esses algoritmos costumam ter maior complexidade computacional do que os métodos do tipo *filter*. O procedimento consiste em classificar "bons" atributos através de um algoritmo de classificação. Porém com esses métodos há um risco de ocorrer *overfitting* dos dados, que é quando o modelo adotado não tem a capacidade de se adaptar a outros conjuntos de dados. Isto pode ocorrer quando a quantidade de objetos for insuficiente.

Os métodos *embedded* realizam uma junção dos métodos *filter* e *wrapper*, onde é utilizado a abordagem *filter* para realizar algum tipo de pré-processamento. Em seguida os atributos resultantes são avaliados pelo algoritmo de aprendizagem. Dos métodos utilizados neste trabalho, os métodos SPEC [6] e Laplacian Score [5] são do tipo *filter*. O método iDetect [7] é do tipo *wrapper* e o método GLSPFS [8] é do tipo *embedded*.

Os métodos Laplacian Score (LS) e SPEC são escolhas clássicas na literatura de seleção de features. Já o iDetect e o GLSPFS são métodos mais recentes, que se propõem a conseguir melhores resultados que os métodos clássicos. Nossa

escolha de limitar os experimentos a essa coleção de métodos, foi portanto, por um lado, motivada por contar com representantes muito usados (clássicos e mais recentes) de cada uma das categorias *filter*, *wrapper* e *embedded* e, por outro lado, foi também resultante da impossibilidade de executar outros métodos não-supervisionados, também propostos na literatura, em conjuntos de dados com grandes números de atributos, utilizando-se de um computador convencional. Isso nos fez descartar, neste trabalho, a utilização de métodos como MCFS [9], UDFS [10], FSASL [11], JELSR [12] e Sparse K-Means [13]. Acreditamos, em trabalhos futuros, que a construção de implementações otimizadas desses métodos, que explorem processamento paralelo/distribuído (potencialmente utilizando GPUs) possa viabilizar sua utilização em conjuntos de dados como os testados.

A seguir serão explicados, de forma resumida, os conceitos e os parâmetros de funcionamento dos métodos que serão utilizados neste trabalho.

2.1 IDETECT

O algoritmo iDetect, consiste em detectar estruturas, de forma iterativa, em conjunto de dados de alta dimensionalidade. A essência está na definição de um critério para quantificar a presença de uma estrutura de dados [7]. A Figura 1 mostra as estruturas de quatro conjuntos de dados. Enquanto os quatro primeiros exemplos são mais fáceis de identificar suas estruturas, o quinto possui muito ruído nos dados, tornando-se mais complexa a identificação de algum padrão.

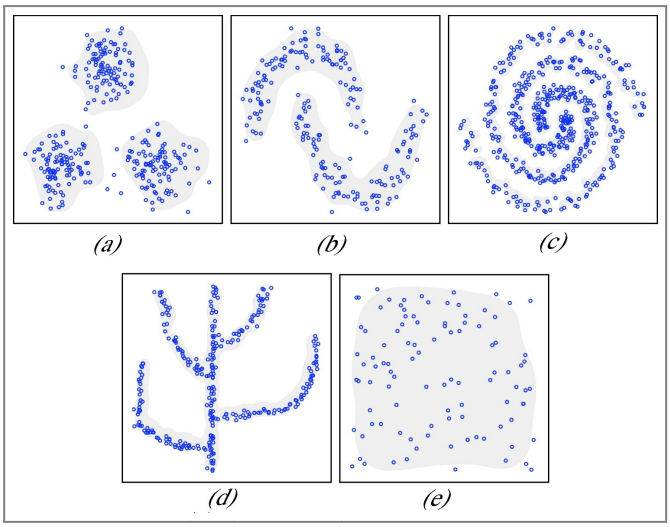


Figura 1. Exemplos de Possíveis Estruturas Presentes em Conjunto de Dados

Fonte: [7]

O iDetect pressupõe que para encontrar uma possível estrutura nos dados é necessário haver homogeneidade dentro do mesmo agrupamento, bem como separação entre os *clusters*. Conforme a Figura 1, é notável nos quatro primeiros exemplos grandes espaços em branco, que realizam essa separação entre os dados. O método não é totalmente livre de parâmetros, para sua execução são necessários quatro parâmetros: o σ (kernel), o λ (parâmetro de regularização), o número de iterações e uma medida de distância que pode ser a euclideana ou Manhattan.

O kernel foi introduzido pelo iDetect para minimizar o problema de comparação da distância entre um ponto e seu vizinho mais próximo. Percebeu-se que o cálculo dessa distância pode ser formulado como um problema de otimização inteira, que é NP-*Hard*. Dessa forma, o problema inicial é relaxado para que as variáveis pudessem ser reais, utilizando o conceito de entropia negativa [14], onde o kernel é responsável por determinar o quão importante um atributo é em um determinado *cluster*. O parâmetro também serve para definir a convergência do algoritmo, assim quanto menor for seu valor mais iterações serão necessárias para a convergência. O kernel não é considerado tão crítico quanto o λ.

Ao se utilizar apenas o parâmetro σ é possível obter atributos com scores muito próximos a zero, prejudicando o algoritmo quando o mesmo é utilizado em conjuntos de dados de alta dimensionalidade. Para solucionar este problema se utilizou uma penalização sobre os scores obtidos [15], surgindo assim o parâmetro λ, que estabelece um controle sobre a dispersão dos dados. O parâmetro, chamado de parâmetro de regularização, é também um critério de parada do algoritmo. Devido à sua criticidade, o λ pode ser definido através do GAP estatístico [16].

Nos experimentos realizados no trabalho que define o iDetect [7], observou-se que sua execução se deu em alguns segundos, mesmo para conjuntos onde o número de variáveis era maior que 5K, mesmo que sua complexidade de pior caso seja O(N^2J), onde N é número de instâncias e J a dimensionalidade do conjunto de dados.

2.2 LAPLACIAN SCORE

O algoritmo tem a premissa de que uma estrutura local em um espaço de dados é mais importante do que uma estrutura global [5]. O método tem como base um grafo construído a partir das aproximações entre os objetos. O método possui o parâmetro W como entrada, que se refere a uma matriz de similaridade entre os objetos. A pontuação dos atributos é de acordo com sua preservação em uma estrutura local, também denominado como Locality Preserving Projections (LPP) [17].

Considere que tenhamos um matriz $n \times m$, onde n se refere aos objetos e m os atributos. Após calcularmos a similaridade entre os objetos a partir de uma medida de distância, como a euclideana, podemos identificar os objetos mais similares, possibilitando a construção de um grafo, onde um objeto i, representado como um vértice, pode estar conectado ao objeto j. A partir deste grafo é construído uma matriz de adjacência $W \in \mathbb{R}^{n \times n}$, e ao realizarmos a ligação entre os objetos i e j, atribuímos um peso chamado k_{ij}, que representa o par (i, j) na matriz W. Com a matriz W também podemos calcular a densidade de cada objeto sobre sua vizinhança. Considere uma matriz D definida por: $D_{ij} = d_i$, se $i = j$, e $D_{ij} = 0$ caso contrário, onde $d_i = \sum_{j=1}^{n} k_{ij}$, assim quanto mais próximos os

objetos estiverem do objeto i, maior será o valor de d_i. Com as matrizes W e D, podemos calcular a matriz de Laplace L definida por:

$$\mathrm{L} = D - W \tag{1}$$

Com as matrizes L, D e vetor 1 (vetor com todos os seus elementos sendo 1), podemos calcular o score de Laplace (LS) atribuído a um atributo f da seguinte forma:

$$\mathrm{LS(f)} = \frac{\tilde{f}^T L \tilde{f}}{\tilde{f}^T D \tilde{f}}, onde \tilde{f} = f - \frac{f^T D 1}{I^T D 1} 1, \tag{2}$$

Para a execução do método LS são necessários dois parâmetros, o knn-size e o weight mode. O parâmetro knn-size se refere ao k utilizado pelo algoritmo knn para a construção do grafo. Desta forma, após calcular as distâncias entre os objetos, a partir de uma métrica de similaridade, realizamos a conexão entre dois objetos se eles estiverem entre os k vizinhos mais próximos.

O parâmetro weight mode possui duas opções, binário e o heat kernel, para calcular o peso que será atribuído sobre as arestas. O modo binário, o modo mais simples, atribui 1 como peso para as arestas dos objetos que estiverem conectados. O heat kernel, também conhecido como RBF kernel, trata-se de uma popular medida de similaridade, onde o valor atribuído para a conexão entre os objetos i e j, pode ser definido da seguinte forma:

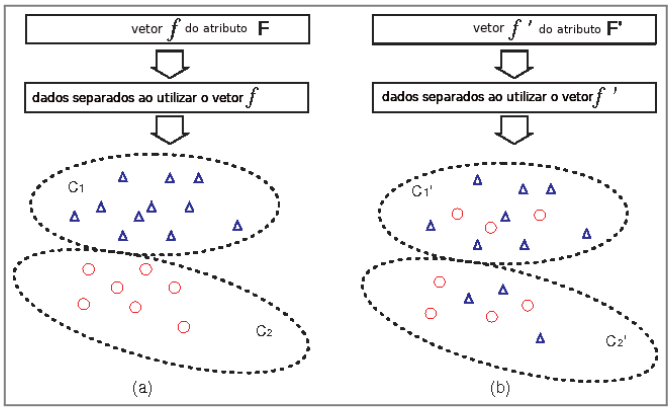
$$\mathrm{S_{ij}} = e^{-\frac{||x_1 - x_2||^2}{(2)}} \tag{3}$$

Onde $||x_1 - x_2||^2$ se refere a distância euclideana quadrática entre os dois vetores de variáveis dos objetos i e j. Em seguida será descrito o método SPEC, que trata-se de uma extensão ao método Laplacian Score.

2.3 SPEC

O SPEC propõe uma solução de seleção em um cenário supervisionado e também não-supervisionado. Ele é baseado na teoria espectral de grafos [18]. O grafo, para o modo supervisionado, consiste na ligação entre objetos de uma mesma classe. Para o caso não-supervisionado o grafo é construído a partir de uma matriz de similaridade entre os objetos, conforme definido pelo método Laplacian Score. A ideia é obter padrões a partir da formação de possíveis estruturas (agrupamentos) presentes no grafo, através de seu espectro, de forma a atribuir scores para os atributos que favoreçam a formação de estruturas mais consistentes.

A Figura 2 mostra os possíveis *clusters* formados a partir das informações de estruturas obtidas pelo espectro do grafo. Onde é possível observar que a variável, aqui representada por F (*feature*), é mais relevante para a definição de clusters do que a F'.



Fonte: [6]

Figura 2. Clusters Formados pelo Espectro de Grafos

O SPEC possui dois parâmetros, a matriz W e o *style*. A matriz W é a mesma utilizada pelo método LS mostrado anteriormente, o parâmetro *style* se refere aos autovalores utilizados pelo espectro do grafo, gerado pelo W. Considere uma matriz de adjacência M, gerada por um grafo G (construído a partir da matriz W), definimos o polinômio característico dessa matriz como:

$$\mathrm{pG}(\lambda) = det(\lambda I - M) \tag{4}$$

Onde *det* se refere ao valor do determinante de $\lambda I - M$, e I a matriz identidade de M. O λ é considerado como um autovalor do grafo G, quando λ é uma raiz de pG. Se M possui s autovalores distintos, $\lambda 1 > ... > \lambda s$, com multiplicidades iguais, respectivamente, a m($\lambda 1$),..., m(λs), o espectro do grafo G é definido como a matriz $2 \times s$, onde a primeira linha é constituída pelos autovalores distintos de M dispostos em ordem decrescente e a segunda, pelas suas respectivas multiplicidades algébricas [19].

Desta forma, a utilização dos autovalores é realizada a partir do parâmetro *style*, que pode ter uma das seguintes opções:

1. Quando o parâmetro possui valor igual a -1 são utilizados todos os autovalores do grafo. Para calcular o score de uma variável F_1 é utilizada a seguinte equação:

$$\mathrm{Score(F_1)} = \frac{f^T L f}{f^T D f}, (LaplacianScore) \tag{5}$$

   Onde f se refere ao vetor de valores da variável F_1.

2. Quando o parâmetro possui valor igual a 0, utilizam-se todos os autovalores, exceto o primeiro, o cálculo do score é definido por:

$$\mathrm{Score(F_1)} = \frac{\sum_{j=1}^{n-1} \alpha_j^2 \lambda_j}{\sum_{j=1}^{n-1} \alpha_j^2} \tag{6}$$

Onde $\alpha = cos\theta_j$ e θ é o ângulo entre f e o autovetor ε_i $(0 \leq i \leq n-1)$ obtido a partir dos autovalores do espectro do grafo.

3. A terceira opção do parâmetro é a partir de um valor predefinido k, onde $k \geq 2$, que visa utilizar os primeiros k autovalores, exceto o primeiro. O cálculo para o score nesta opção é definido a seguir:

$$\text{Score}(\text{F}_1) = \sum_{j=1}^{n-1} (2 - \lambda_j)\alpha_j^2 \tag{7}$$

Observe através do parâmetro *style*, que quando utilizamos a primeira opção (-1), o score atribuído são os mesmos quando utilizamos o método LS, e que a terceira opção para este parâmetro exige a necessidade de um outro parâmetro (k), desta forma não será utilizado essa opção.

O método SPEC segue a teoria que uma variável pode ser considerada consistente caso atribua valores similares aos objetos vizinhos, de forma a preservar os autovalores e a estrutura gerada a partir do grafo. Assim, as variáveis que realizem modificações sobre a estrutura formada inicialmente pelo grafo, tendem a ter scores mais baixos. Observe que tanto o método LS quanto o SPEC realizam uma validação local para cada atributo.

2.4 GLSPFS

O GLSPFS [8] é um método capaz de realizar a seleção em um cenário supervisionado, não-supervisionado e também semi-supervisionado (realiza a seleção ao utilizar dados com e sem as informações de classe). O método busca preservar as informações intrínsecas (estruturas) presentes em conjuntos de dados de alta dimensionalidade [20], esta preservação é vista de forma local e global. Porém para o cenário não-supervisionado, o método irá focar em preservar a estrutura local. O método obtém as informações sobre a geometria local dos dados a partir da utilização de três algoritmos: local linear embedding (LLE) [21], locality preserve projection (LPP) [17] e local tangent space alignment (LTSA) [22]. Neste trabalho foram utilizadas todas as três possibilidades para o método. Além do parâmetro referente ao algoritmo utilizado para a análise local das *Features*, também é necessário fornecer ao método a matriz W, como também ocorre nos métodos LS e SPEC. Através da Tabela 1 é possível observar as parametrizações necessárias para o método GLSPFS, assim como para os demais métodos.

3. A Metodologia de Seleção

Este trabalho se refere a uma metodologia e não apenas um método de seleção de *Features*, pois busca apresentar estratégias (um guia) para solucionar problemas, considerados em aberto na literatura, para a seleção de atributos em conjuntos de dados de modo não-supervisionado, onde essas estratégias são independentes de métodos, tornando a metodologia adaptável ao uso de quaisquer métodos.

A metodologia será capaz de efetuar a redução da dimensionalidade dos dados, a partir da seleção de atributos em um cenário não supervisionado, considerando que o conjunto de variáveis resultante seja capaz de gerar os 'melhores' grupos, a partir de um algoritmo de *Clustering*. Neste trabalho optaremos, inicialmente, pela execução de um K-Means clássico, tal escolha se deu por sua popularidade entre os algoritmos de *Clustering*, porém em um trabalho futuro poderá ser realizada uma investigação, bem como a adoção de outros métodos de agrupamento.

Antes de descrevermos o fluxo da metodologia serão abordados alguns requisitos que foram estabelecidos. No decorrer deste trabalho, utilizaremos os termos entre colchetes para melhor referenciá-los:

(1) A metodologia precisa ser independente de métodos de seleção de atributos, ou seja, deve ser possível a fácil inclusão/remoção dos algoritmos para pontuar os atributos, dessa forma é possível utilizar diversas abordagens. Na medida em que novos métodos sejam propostos na literatura e/ou os métodos até então utilizados nesta metodologia se tornarem obsoletos, a fácil adição/remoção proporcionará a obtenção de melhores performances, ou a otimização dos métodos já existentes. [INDEPENDÊNCIA].

(2) O processo de seleção de atributos precisa ser orientado por índices não supervisionados, não havendo a exigência da informação da classe para os objetos no conjunto de dados utilizado, assim poderemos garantir que a metodologia seja aplicável em um cenário totalmente não-supervisionado. [NÃO-SUPERVISIONADO].

(3) A metodologia precisa estar apta para utilizar, em paralelo, múltiplos métodos para realizar a seleção, possibilitando, inclusive, a junção entre eles, com o objetivo de obter os melhores resultados. Este requisito possibilita diversificar a seleção das variáveis, pois o resultado de um método pode ser complementar a outro. [MÚLTIPLO].

Para atender o requisito de [INDEPENDÊNCIA], assumimos como única exigência para cada método a ser utilizado, que ele atribua uma espécie de pontuação sobre as variáveis. Já para o requisito [NÃO-SUPERVISIONADO] pressupõe-se que seja utilizada uma métrica que não exija a informação da classe dos objetos, o que será detalhado na seção 3.4. Finalmente, para o requisito [MÚLTIPLO], a seção 3.2 definirá uma abordagem para a utilização de múltiplas configurações em cada método e na seção 3.5 a estratégia utilizada para a combinação dos resultados. Ao final, será disponibilizada uma lista de opções para que o especialista de dados possa analisá-las de maneira a definir o que parece ser o "melhor"conjunto de atributos.

Para facilitar a referência à metodologia proposta neste trabalho, utilizaremos as iniciais, em inglês, de cada requisito

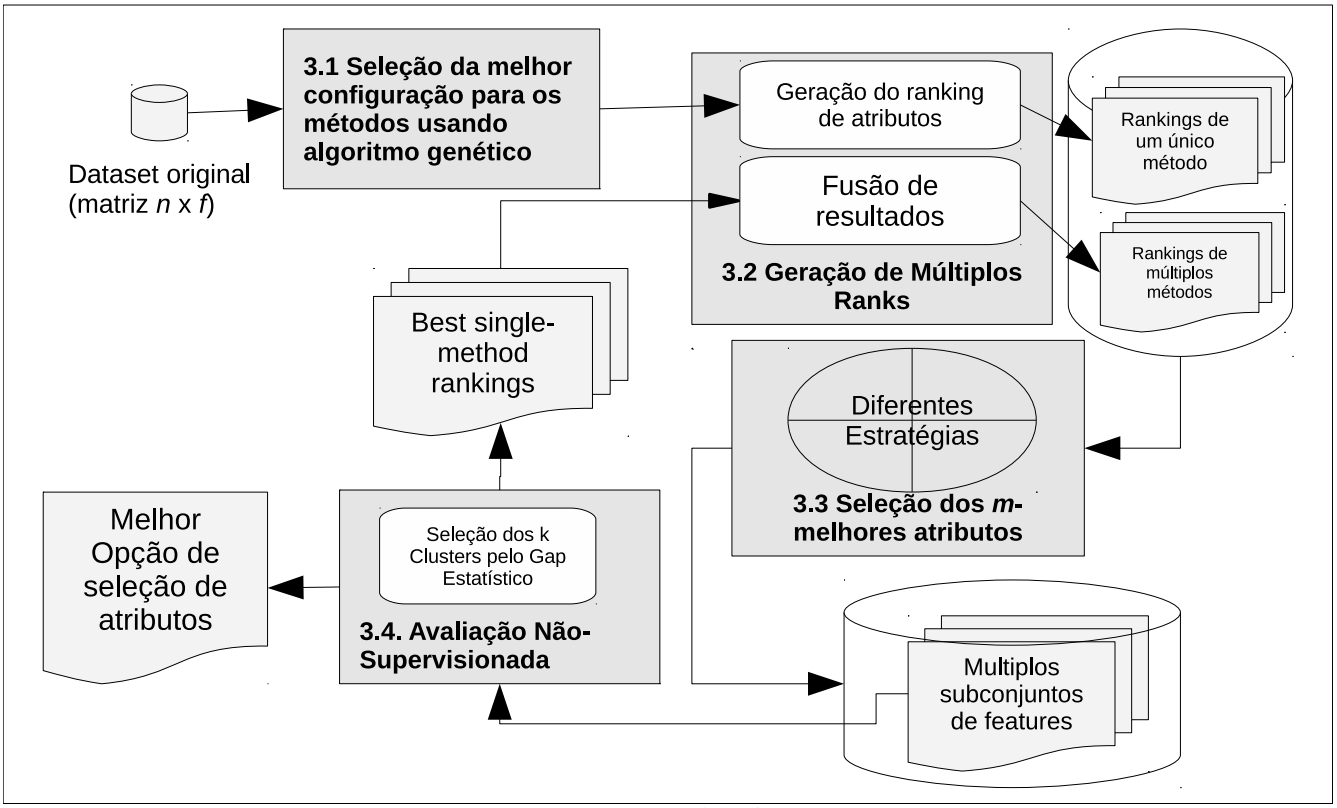


Figura 3. A Metodologia MUI

e a chamaremos de MUI (*Multiple*, *Unsupervised* e *Independent*). A metodologia MUI pode ser caracterizada como um conjunto de estratégias para realizar a seleção de *Features*, tal conjunto pode ser compreendido como uma abordagem *meta-embedded*, onde é possível combinar múltiplos tipos de métodos, *wrapper* e *filter*, com um algoritmo de *Clustering* baseado em um critério de otimização, com o objetivo de disponibilizar uma lista com as melhores opções. A metodologia MUI também pode ser caracterizada como uma abordagem do tipo *ensemble* [23], capaz de otimizar os parâmetros dos métodos, além de tratar questões não tratadas pelos métodos (ex.: seleção das m-melhores *Features*). A Figura 3 ilustra o fluxo de funcionamento do MUI. Espera-se que ao final, o MUI seja capaz de fornecer boas configurações, realizando a seleção, de modo a revelar agrupamentos, considerados relevantes, para os especialistas de dados. As etapas do MUI serão detalhadas a seguir, bem como algumas estratégias que visam solucionar alguns questionamentos levantados na introdução deste trabalho, tais como uma abordagem para selecionar as m-melhores *Features* e como realizar a combinação de resultados dos método de seleção.

3.1 Normalização dos Dados

Em muitos conjuntos de dados é necessário realizar uma normalização para que os dados sejam comparáveis, ou seja, é importante que os dados estejam em uma mesma escala, sendo uma importante etapa de pré-processamento de dados [24].

Considerando como entrada uma matriz de n (objetos) $\times$ f (atributos originais), após a normalização a saída será um novo conjunto de dados $n \times f$, onde cada coluna será normalizada para que os valores estejam em uma mesma escala, por exemplo entre 0 e 1. Nos experimentos desse trabalho o método utilizado para esta tarefa foi o *straighforward normalization*, também utilizado no trabalho em [7] (no entanto, qualquer outra forma de normalização mais adequada para o conjunto de dados em questão pode ser utilizada). A Equação 8 detalha o processo de normalização. Onde cada atributo f e objetos i são representados em uma matriz de dados $n \times f$.

$$x_{if_1} = \frac{x_{if} - \min_{j=1}^{n} x_{jf}}{\max_{j=1}^{n} x_{jf} - \min_{j=1}^{n} x_{jf}} \tag{8}$$

Após esta etapa os dados seguirão normalizados para as etapas subsequentes. A seguir será apresentada uma abordagem para tentar solucionar a escolha dos parâmetros para a execução dos métodos de seleção.

3.2 Seleção de Parâmetros (Questão Q4)

Considerando que a execução do MUI seja de um modo totalmente não supervisionado, isto é sem uma intervenção manual, é necessário configurar os métodos de seleção, pois eles não são totalmente livres de parâmetros. A Tabela 1 mostra os parâmetros utilizados para alguns métodos na literatura (cf. [11]).

Tabela 1. Total de Configurações por Método

Método	Parâmetros	Valores	Configurações
LS	matriz W	knn-size = [5,6,7,8] weight mode = ['binary', heat kernel]	8
SPEC	matriz W	knn-size = [5,6,7,8] weight mode = ['binary', heat kernel]	16
	style	[-1, 0]	
GLSPFS	local-type	['LPP', 'LLE', 'LTSA']	12
	matriz W	knn-size = [5,6,7,8] kernel = *'Gaussian'*	
iDetect	sigma	$[10^{-5}, 10^{-3}, 10^{-1}]$	30
	lambda	$[2^1, 2^3, 2^5, 2^7, 2^9]$	
	distance	['euclidean', 'block']	
	iterations	20	

Para guiar a seleção não supervisionada dos parâmetros, utilizaremos neste trabalho a hipótese de que uma "boa" configuração é aquela que consiga gerar uma maior variância no ranking de *Features*. Tal hipótese é baseada na ideia de que quando a variância dos scores é pequena, significa que o método, assim configurado, não está conseguindo distinguir bem os atributos, dificultando a eliminação dos atributos considerados redundantes (possuem um score similar). Tal critério poderia ser revisto em um trabalho futuro, podendo ser substituído por um indicador de avaliação de clusters.

Para otimizar os parâmetros de modo a gerar rankings com maior variância nos scores, utilizamos neste trabalho a abordagem de um algoritmo genético clássico. A seguir são descritas as etapas do algoritmo:

(1) É definida uma população inicial com as 50 primeiras configurações, escolhidas de forma aleatória;

(2) É realizada uma competição, entre os indivíduos da população inicial, onde são selecionadas as 10 melhores configurações;

(3) Essas 10 configurações são armazenadas para a próxima geração;

(4) A partir dessas 10 configurações são gerados 10 novos indivíduos através do processo de *crossover* e mais 10 novos indivíduos a partir do processo de mutação;

(5) Esses 30 indivíduos (10 selecionados da população individual, 10 do processo de *crossover* e 10 do processo de mutação) são colocados em uma nova competição que resultarão 10 indivíduos;

(6) As etapas anteriores se repetem por 5 gerações (competições);

Ao término destas competições teremos uma coleção de 10 rankings, em seguida aplicaremos um método para a escolha das *m*-melhores *Features* em apenas um ranking (aquele que tiver a maior variância entre os demais), o que corresponde à questão Q3 levantada na introdução deste trabalho.

3.3 Seleção do Número de Atributos (Questão Q3)

Considerando que tenhamos um ranking de variáveis, gerado a partir de scores decrescentes atribuídos pelos métodos de seleção, é necessário definirmos um ponto de corte *m*, que selecione o conjunto de variáveis a ser mantido. Neste trabalho propomos a escolha deste número *m* através do ponto de inflexão. Este método considera que possamos ordenar os atributos de acordo com os scores, de forma descendente, e traçar uma linha para representar esses valores. Foi observado que esta linha tem pontos de inflexão que podem ser bons candidatos para o *m*. Após esses pontos foi observado que os scores tendem a decrescer de forma mais lenta.

Podemos calcular o ponto que corresponde a "maior" inflexão ao calcular a segunda derivada de cada valor em cada ponto no gráfico da linha e obter o número de variáveis correspondente com a maior segunda derivada. A Figura 4 simula um exemplo de scores para 15 variáveis (*Features*), onde seriam escolhidas as 8 melhores *Features*. Faremos isto utilizando a expressão 9:

$$\max_{f=2}^{F-1} s(f+1) + s(f-1) - 2 * s(f) \tag{9}$$

Onde F é o total de números *Features* e $s(f)$ representa o score da *Feature* no Index f da lista ordenada, de forma decrescente, de *Features* por Score.

A seguir será abordada a métrica utilizada para avaliar o conjunto obtido pelas *m*-melhores *Features*.

3.4 Avaliação do Conjunto de *Features*

Neste trabalho consideramos a seleção de *Features* em atividades de *Clustering*, assim utilizaremos a abordagem clássica do algoritmo do *k*-Means, como implementado em [3]. A escolha do *k*-Means, neste trabalho, é devido à sua fácil execução e por ser um dos mais populares algoritmos para *Clustering*, porém o mui, em um trabalho futuro, poderia fazer uso de outros algoritmos, bem como uma combinação de algoritmos de *Clustering*. A execução do *k*-Means, por sua vez necessita do número de *Clusters* desejado como parâmetro e uma métrica

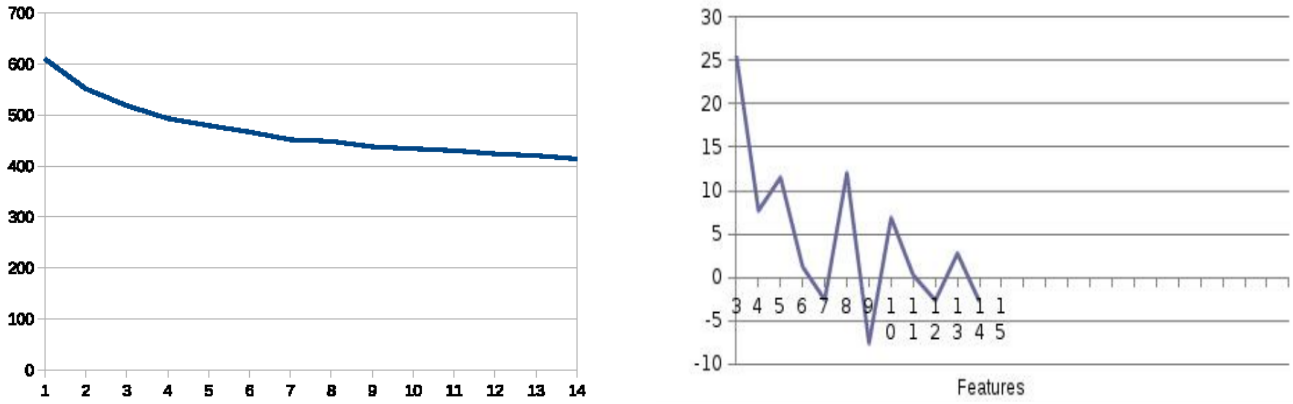


Figura 4. Simulação de Scores para as *Features*

de distância, que neste trabalho utilizaremos como padrão a distância *Euclideana* para a formação dos *Clusters*. Como consideramos um contexto totalmente não-supervisionado (ou seja, não é conhecido o número de classes desejado), neste trabalho escolheremos esse parâmetro através da estratégia do Gap estatístico [16].

A medida utilizada para avaliar o conjunto de *Features*, gerado na etapa anterior, será baseada na silhueta, mais especificamente, a média das silhuetas de todos os objetos. Após as iterações do *k*-means, buscaremos obter agrupamentos mais coesos (homogeneidade) e separados (heterogeneidade). A silhueta é uma métrica não-supervisionada que pode ser definida da seguinte forma: seja a(i) a média das distâncias entre o objeto i e os demais objetos pertencentes ao mesmo *Cluster* que i, e b(i) a média das distâncias de i com os objetos dos demais *Clusters*, a silhueta do objeto i, s(i), é definida da seguinte forma:

$$\mathrm{s(i)} = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)} \tag{10}$$

Será considerado o "melhor" conjunto de *Features* aquele que obtiver o maior valor para a silhueta, ou seja, após termos um ranking como entrada, é calculado o valor da silhueta para o conjunto gerado a partir do ponto de inflexão, em seguida é selecionado e armazenado o conjunto que teve a melhor silhueta (maior). A seguir será abordado uma estratégia para combinar os rankings dos métodos de seleção de *Features*. A ideia é verificar se essa combinação de resultados poderá aumentar a performance e disponibilizar mais opções de escolha para o especialista de dados.

3.5 Combinação de Resultados (Questão Q5)

A combinação dos resultados dos métodos será através da contagem de Borda[25], o método se trata de um mecanismo de votação utilizado para gerar um ranking global de candidatos (neste trabalho, *Features*) a partir de rankings individuais dos eleitores (neste trabalho, cada ranking obtido por cada configuração dos métodos de seleção de *Features*). Suponha que tenhamos três *Features*, feature-a, feature-b, feature-c, e uma configuração de um método de seleção de *Features*, A. Portanto podemos ter o seguinte ranking de *Features* A = [feature-b, feature-a, feature-c] (isto significa que a feature-b obteve um Score melhor do que feature-a e feature-c). Obtendo um segundo ranking de *Features*, B, tem-se B = [feature-a, feature-c, feature-b]. Através da contagem de Borda, cada *Feature* obtém um número de pontos correspondente a sua posição em cada ranking. Assim, a feature-a tem 2 + 1 = 3 pontos, feature-b tem 1 + 3 = 4 pontos e feature-c 3 + 2 = 5 pontos, logo o resultado obtido pela junção dos resultados através da contagem de Borda é: C = [feature-a (3 pontos), feature-b (4 pontos), feature-c (5 pontos)].

De modo a limitar o número de possíveis combinações, será realizada a junção entre os melhores resultados (rankings que obterem o maior valor de silhueta) para cada método. Este trabalho toma como base 4 métodos, desta forma são 15 possíveis opções, isso quando somados os resultados individuais de cada método mais as possíveis combinações. A combinação dos métodos possibilita explorar os dados de diferentes formas, como por exemplo, a abordagem para rankeamento das *Features* do método iDetect é diferente do SPEC, e eles podem se complementar.

Para realizar a combinação de resultados será selecionado o número de *Features* a partir de "números mágicos", tais como as 10, 20, 30 ... 100 *Features* mais bem posicionadas no ranking, pois a abordagem pelo ponto de inflexão necessita que cada *Feature* tenha um score atribuído, enquanto que o ranking gerado pela combinação de resultados não possui scores. Em seguida será avaliado se essas premissas serão atendidas através dos experimentos realizados.

4. Experimentos e Avaliação dos Resultados

Nesta seção serão verificados os resultados obtidos pela metodologia em 11 diferentes conjuntos de dados. Esses *Datasets* estão publicamente disponíveis através do repositório "scikit-feature"[2]. A ideia é verificar o comportamento da metodologia em conjuntos de diferentes domínios (ex.: Bioinformática, imagens e texto), sendo três *Datasets* da área de

imagens, cinco de bioinformática e três de texto. A Tabela 2 mostra as características desses conjuntos de dados. Todos os conjuntos compartilham uma característica em comum, que é a de possuir um número elevado de *Features*.

Nos experimentos serão utilizados os índices supervisionados: *Normalized Mutual Information* (NMI) [26] e o *Corrected Rand* (CR) [27], para avaliarmos os grupos gerados, quando comparados com os *Labels* dos dados. Nossa análise inicial parte da verificação do melhor resultado obtido pelos métodos ao utilizarmos uma configuração *default* (essa configuração é gerada automaticamente pelo método quando os valores dos parâmetros não são informados). Para a geração dos resultados desta análise inicial será utilizado o *k*-means como método de geração dos *Clusters*, com o número de *Clusters* igual ao número de *Labels* e os conjuntos de *Features* serão gerados a partir de números mágicos. A Figura 5 mostra a performance do método LS, utilizando uma configuração padrão, no conjunto de dados WARPAR10P.

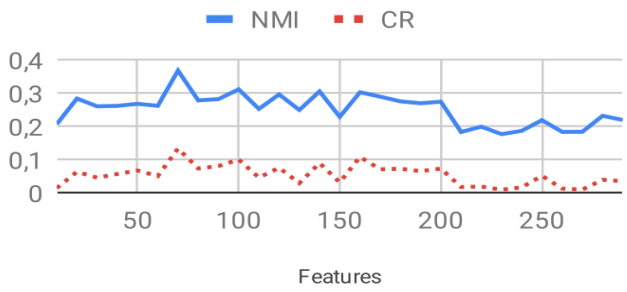


Figura 5. Resultado do Método LS no *Dataset* WARPAR10P

Observe que é necessário definir um ponto de corte manual para selecionar o número de *Features*, assim serão selecionadas as *m Features* que alcançarem o maior valor para os índices de NMI e CR. Neste exemplo, o método LS obteve a melhor performance ao utilizar as 70 *Features* mais bem posicionadas no ranking de scores. Consideramos esse desempenho como nossa base de comparação, pois esses resultados mostram o melhor resultado que pode ser obtido nos métodos de seleção, utilizando seus parâmetros padrão. Também será comparado o desempenho obtido pelo MUI com os resultados obtidos ao se utilizar todas as *Features*.

A metodologia MUI disponibilizará ao especialista de dados uma lista com 15 opções, sendo 4 opções referentes aos resultados obtidos pelos métodos quando executados de forma isolada e 11 quando realizamos a combinação de resultados entre os métodos, através da contagem de Borda. Para melhor referenciar os resultados, será utilizada a letra inicial de cada método e o hífen quando se referir a uma combinação de resultados. A Tabela 3 mostra os resultados obtidos, para essas 15 opções, ao executarmos a metodologia no conjunto de dados WARPAR10P.

Observe os resultados do MUI, através da Tabela 3, que entre as opções disponíveis o método GLSPFS (opção 4) obteve a maior silhueta (0,39479), alcançando um NMI de 0,31698 e CR de 0,05974 ao utilizar apenas 6 *Features*, selecionadas pelo ponto de inflexão (INF), porém há uma outra opção, opção 2, através do método SPEC (S), que foi capaz de obter um NMI de 0,51821 e CR de 0,21137, selecionando 6 *Features* também pelo ponto de inflexão. Nota-se que esses resultados, produzidos com configurações geradas de forma totalmente não supervisionada, possuem opções com bons valores para os índices supervisionados, NMI e CR.

Ao compararmos os melhores resultados dos métodos, quando utilizamos uma configuração *default*, com aquela que seria a melhor opção do MUI, ambos utilizamos o maior NMI e CR, podemos construir a Tabela 4. Consideramos essa Tabela como sendo nosso benchmark ao compararmos o desempenho do MUI em relação aos métodos presentes na literatura, utilizados neste trabalho.

Observe que o MUI obteve um desempenho melhor, em 8 dos 11 *Datasets* avaliados, e que a combinação de resultados, através da contagem de Borda, forneceu também bons resultados. Também é possível observar que o método LS não aparece, em nenhum momento, como sendo uma boa opção de seleção, podemos considerar em um trabalho futuro a substituição/remoção deste método. A seguir serão apresentadas as considerações finais e os trabalhos futuros a serem realizados.

5. Conclusões e Trabalhos Futuros

Através dos experimentos realizados, podemos observar que os métodos de seleção de *Features*, utilizados na literatura e também neste trabalho, não se mostraram equivalentes quando comparados pelos índices NMI e CR. Além disso, não houve um método consistentemente melhor nos diferentes conjuntos de dados utilizados, respondendo assim as questões Q1 e Q2 levantadas na introdução deste trabalho. Também é possível notar que o MUI tem como principal vantagem uma busca mais eficiente por uma boa configuração, a partir dos métodos utilizados, de maneira não-supervisionada, utilizando-se da silhueta dos *Clusters* gerados. Essa busca (heurística) é computacionalmente de baixo custo na implementação utilizada, com o uso de um algoritmo genético clássico. Uma outra vantagem ao utilizar o MUI é que podemos alterar a coleção de métodos utilizados, mantendo a mesma metodologia de seleção e otimização, com resultados guiados por critérios não-supervisionados, tornando a execução da metodologia totalmente automática, isto é, sem a necessidade de uma intervenção manual, mesmo que a escolha final do conjunto de atributos seja feita por um especialista.

Também foi possível observar que o critério não-supervisionado utilizado, a silhueta, nem sempre correspondeu aos maiores índices supervisionados (NMI e CR). Note que nem sempre os agrupamentos naturais dos dados refletem as classes previamente conhecidas, podendo revelar outros padrões, de forma que baixos índices supervisionados não necessariamente indicam um resultado "ruim". Dito isto, observamos que na maioria dos casos, o MUI obteve bons resultados nos índices supervisionados. Ou seja, entre as opções disponibilizadas pelo MUI (selecionadas de forma não

Tabela 2. Conjuntos de Dados

Dataset	Domínio	Features	Instâncias	Classes
ALLAML	Microarray	7129	72	2
CARCINOM	Microarray	9182	174	11
PROSTATE-GE	Microarray	5966	102	2
SMK-CAN	Microarray	19993	187	2
TOX171	Microarray	5748	171	4
PIXRAW10P	Imagem	10000	100	10
WARPAR10P	Imagem	2400	130	10
WARPPIE10P	Imagem	2420	210	10
PCMAC	Texto	3289	1943	2
BASEHOCK	Texto	4862	1993	2
RELATHE	Texto	1427	4322	2

Tabela 3. Opções do MUI no *Dataset* WARPAR10P

Opção	Método	Seleção	*Features*	Silhueta	NMI	CR
1	L	INF	5	0,27360	0,25565	0,02311
2	S	INF	6	0,28260	**0,51821**	**0,21137**
3	i	INF	8	0,25267	0,33868	0,06824
4	G	INF	6	0,39479	0,31698	0,05974
5	L-S	M	20	0,22074	0,39569	0,09262
6	L-i	M	10	0,26406	0,28153	0,03110
7	L-S-i	M	20	0,19775	0,37025	0,10598
8	S-i	M	20	0,33214	0,41529	0,13425
9	L-G	M	10	0,17330	0,32755	0,06627
10	S-G	M	10	0,21701	0,51424	0,19194
11	i-G	M	20	0,33645	0,45133	0,16476
12	L-S-G	M	10	0,20036	0,45650	0,11599
13	L-i-G	M	10	0,20625	0,25281	0,02112
14	S-i-G	M	10	0,28732	0,35325	0,10151
15	L-S-i-G	M	10	0,18808	0,30709	0,06973

supervisionada), temos configurações que atingem valores para os índices supervisionados melhores que aqueles obtidos pelos métodos em configurações *default*.

Recapitulando as questões levantadas na introdução e as soluções exploradas neste trabalho, em busca de suas respostas, foi possível obter os seguintes resultados:

- (Q1) e (Q2): Através dos experimentos, verificamos que os métodos obtiveram resultados diversos, portanto nenhum deles foi equivalente a outro. Também não houve método que sempre obtivesse resultados melhores, reforçando o interesse ao especialista em explorar as diferentes opções de métodos de seleção de *Features* disponíveis, sem considerar qualquer um deles como a solução padrão para qualquer problema.

- (Q3) e (Q5): Quando os métodos de seleção são executados individualmente, propusemos o ponto de inflexão dos scores das features para selecionar o ponto de corte (*m*-melhores). Na fusão de resultados de diferentes métodos, utilizamos o método de Borda para criação de um ranking global e depois selecionamos o ponto de corte através de "números mágicos" guiados pela melhor silhueta. Essa abordagem obteve bons resultados nos experimentos.

- (Q4): Propusemos utilizar a maximização da variância dos scores obtidos como objetivo, e neste trabalho utilizamos um algoritmo genético clássico para tal.

Portanto, com o uso da metodologia foi possível descobrir configurações, de forma não supervisionada, que possuíam uma concordância razoável com as classes previamente conhecidas dos conjuntos de dados utilizados. Assim podemos considerar, ao final, que a metodologia proposta (MUI) obteve resultados satisfatórios. Entretanto, é possível desde já visualizar melhorias que podem ser exploradas em trabalhos futuros.

Acreditamos que algumas melhorias podem ser feitas no fluxo de funcionamento da metodologia. Uma dessas melhorias se refere ao critério adotado para a escolha dos conjuntos de *Features*. Neste trabalho foi utilizada a silhueta dos *Clusters* como métrica de avaliação, porém outros critérios poderiam ser utilizados, tais como o *Dunn Index* [28], po-

Tabela 4. Resultados

Dataset	ALL		LS				SPEC				iDetect				GLSPFS				Best MUI			
	NMI	CR	FEA	NMI	FEA	CR	FEA	NMI	FEA	CR	FEA	NMI	FEA	CR	FEA	NMI	FEA	CR	OPT	FEA	NMI	CR
WARPAR10P	0,18329	0,02626	70	0,36915	70	0,13474	60	0,40737	280	0,13960	260	0,44485	260	0,20930	110	0,18127	110	0,03058	**S**	**6**	**0,51821**	**0,21137**
WARPPIE10P	0,32143	0,09848	30	0,23740	20	0,09133	140	0,53736	140	0,28459	180	0,37268	130	0,16123	80	0,34135	180	0,10019	**S-G**	**20**	**0,63126**	**0,34479**
PIXRAW10P	0,94960	0,88802	280	0,62360	280	0,33871	280	0,60410	280	0,36275	160	0,86183	160	0,73587	**250**	**0,94268**	**250**	**0,89241**	S-i-G	200	0,84665	0,65909
TOX171	0,26377	0,26377	210	0,04773	210	0,02860	110	0,21158	230	0,13562	**260**	**0,29630**	**260**	**0,22538**	180	0,06893	200	0,03314	S-i	260	0,29281	0,09343
ALLAML	0,17135	0,23795	120	0,06843	180	0,07323	**290**	**0,63191**	**290**	**0,68475**	100	0,18095	100	0,26459	270	0,16480	270	0,18710	S-G	10	0,32032	0,05361
CARCINOM	0,67127	0,55523	80	0,20063	270	0,06408	280	0,29879	280	0,14983	260	0,67924	260	0,56139	220	0,45118	210	0,30379	**S-i**	**210**	**0,72655**	**0,58676**
SMK-CAN	0,00175	-0,00121	290	0,01308	20	0,01294	120	0,03062	120	0,03455	50	0,03101	50	0,03847	220	0,05898	**220**	**0,06947**	**S-i-G**	**20**	**0,12357**	0,03516
PROSTATE-GE	0,0255	0,02276	20	0,04692	20	0,01515	20	0,07068	20	0,00469	60	0,02393	60	0,02227	110	0,01955	110	0,01603	**S-G**	**10**	**0,14248**	**0,02670**
PCMAC	0.00967	0,00005	260	0,00967	10	0,00076	90	0,00967	10	**0,00076**	190	0,02493	10	0,00004	20	0,00967	10	0,00004	**S-G**	**10**	**0,04811**	0,00029
BASEHOCK	0.00634	-0,00001	40	0,00634	10	0,00003	40	0,00634	10	0,00003	210	0,00584	210	0,00028	10	0,00634	10	0,00003	**i**	**6**	**0,05749**	**0,013546**
RELATHE	0,00128	0,00219	30	0,00672	20	0,00037	30	0,00672	20	0,00037	30	0,01246	30	0,00671	10	0,00672	10	-0,00023	**S**	**46**	**0,10093**	**0,00130**

dendo inclusive surgir a proposta de um novo critério de avaliação. Nesse trabalho seriam realizados estudos para a definição de melhores critérios não supervisionados para a avaliação da qualidade dos *Clusters* formados, podendo levar a combinação de diferentes critérios existentes, buscando encontrar configurações que possuem um perfil equilibrado de qualidade entre diferentes critérios, colocando esse trabalho futuro numa perspectiva multi-critério. Além desse trabalho é possível elencar também os seguintes trabalhos:

(T1) Possibilitar o uso de mais algoritmos de seleção de *Features* no MUI, bem como as devidas combinações, aumentando as possibilidades de obter um resultado final melhor;

(T2) Avaliar o desempenho da metodologia em conjunto de dados com dimensionalidades ainda superiores às utilizadas neste trabalho;

(T3) Realizar um experimento com validação manual por especialistas do domínio, ou com conjuntos de dados que 'visualmente' os resultados revelem informações relevantes, possibilitando validar os padrões descobertos no domínio do conhecimento. Isso poderia ser utilizado como uma forma de treinamento para o MUI, contribuindo para o ajuste da metodologia;

(T4) A metodologia utilizou a execução de um *k*-means clássico como algoritmo de agrupamento para avaliar as *Features* geradas, porém outros algoritmos de agrupamento podem igualmente serem utilizados.

Contribuição dos Autores

Marcos de Souza Oliveira contribuiu no desenvolvimento da ferramenta, na geração dos resultados, na estruturação e escrita do artigo. Sérgio Ricardo de Melo Queiroz contribuiu com apoio no desenvolvimento da ferramenta, elaboração da metodologia para coleta e análise dos resultados, apoio na estruturação, escrita e revisão do artigo.

References

[1] DONOHO, D. L. *et al.* High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, Citeseer, v. 1, p. 32, 2000.

[2] LI, J. *et al.* Feature selection: A data perspective. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 50, n. 6, p. 94:1–94:45, dez. 2017. Disponível em: ⟨http://doi.acm.org/10.1145/3136625⟩.

[3] ARTHUR, D.; VASSILVITSKII, S. k-means++: The advantages of careful seeding. In: SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. [S.l.], 2007. p. 1027–1035.

[4] LIU, H.; MOTODA, H. *Computational methods of feature selection*. [S.l.]: CRC Press, 2007.

[5] HE, X.; CAI, D.; NIYOGI, P. Laplacian score for feature selection. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2006. p. 507–514.

[6] ZHAO, Z.; LIU, H. Spectral feature selection for supervised and unsupervised learning. In: ACM. *Proceedings of the 24th international conference on Machine learning*. [S.l.], 2007. p. 1151–1157.

[7] YAO, J. *et al.* Feature selection for unsupervised learning through local learning. *Pattern Recognition Letters*, Elsevier, v. 53, p. 100–107, 2015.

[8] LIU, X. *et al.* Global and local structure preservation for feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, v. 25, n. 6, p. 1083–1095, 2014.

[9] CAI, D.; ZHANG, C.; HE, X. Unsupervised feature selection for multi-cluster data. In: ACM. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2010. p. 333–342.

[10] YANG, Y. *et al.* l2, 1-norm regularized discriminative feature selection for unsupervised learning. In: *IJCAI proceedings-international joint conference on artificial intelligence*. [S.l.: s.n.], 2011. v. 22, p. 1589.

[11] DU, L.; SHEN, Y.-D. Unsupervised feature selection with adaptive structure learning. In: ACM. *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. [S.l.], 2015. p. 209–218.

[12] HOU, C. *et al.* Feature selection via joint embedding learning and sparse regression. In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2011. v. 22, p. 1324.

[13] WITTEN, D. M.; TIBSHIRANI, R. A framework for feature selection in clustering. *Journal of the American Statistical Association*, Taylor & Francis, v. 105, n. 490, p. 713–726, 2010.

[14] FRIEDMAN, J. H.; MEULMAN, J. J. Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Wiley Online Library, v. 66, n. 4, p. 815–849, 2004.

[15] TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 267–288, 1996.

[16] TIBSHIRANI, R.; WALTHER, G.; HASTIE, T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Wiley Online Library, v. 63, n. 2, p. 411–423, 2001.

[17] HE, X.; NIYOGI, P. Locality preserving projections. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2004. p. 153–160.

[18] CVETKOVIĆ, D. M.; DOOB, M.; SACHS, H. *Spectra of graphs: theory and application*. [S.l.]: Academic Pr, 1980. v. 87.

[19] ABREU, N. M. M. d. *et al.* Introdução à teoria espectral de grafos com aplicações. *Notas em Matemática Aplicada*, v. 27, p. 25, 2007.

[20] GU, Q. *et al.* Joint feature selection and subspace learning. In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2011. v. 22, n. 1, p. 1294.

[21] ROWEIS, S. T.; SAUL, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, American Association for the Advancement of Science, v. 290, n. 5500, p. 2323–2326, 2000.

[22] ZHANG, T. *et al.* Linear local tangent space alignment and application to face recognition. *Neurocomputing*, Elsevier, v. 70, n. 7-9, p. 1547–1553, 2007.

[23] ZHOU, Z.-H. *Ensemble methods: foundations and algorithms*. [S.l.]: Chapman and Hall/CRC, 2012.

[24] SHALABI, L. A.; SHAABAN, Z.; KASASBEH, B. Data mining: A preprocessing engine. *Journal of Computer Science*, v. 2, n. 9, p. 735–739, 2006.

[25] BORDA, J.-C. de. Mémoire sur les élections au scrutin, histoire de l'académie royale des sciences. *Paris, France*, 1781.

[26] STREHL, A.; GHOSH, J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, v. 3, n. Dec, p. 583–617, 2002.

[27] HUBERT, L.; ARABIE, P. Comparing partitions. *Journal of classification*, Springer, v. 2, n. 1, p. 193–218, 1985.

[28] BEZDEK, J. C.; PAL, N. R. Cluster validation with generalized dunn's indices. In: IEEE. *Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*. [S.l.], 1995. p. 190–193.