Pós-Graduação em Ciência da Computação

Marcos de Souza Oliveira

Metodologia de Seleção de Features Não-supervisionada para Clustering em Conjunto de Dados de Alta Dimensionalidade

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife

2018

Marcos de Souza Oliveira

Metodologia de Seleção de Features Não-supervisionada para Clustering em Conjunto de Dados de Alta Dimensionalidade

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Aprendizagem de Máquina e Mineração
Orientador: Sergio Queiroz

Recife
2018

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

O48m Oliveira, Marcos de Souza
Metodologia de seleção de *features* não supervisionada para *clustering* em conjunto de dados de alta dimensionalidade / Marcos de Souza Oliveira. – 2018.
79 f.: il., fig., tab.

Orientador: Sérgio Queiroz.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2018.
Inclui referências e apêndice.

1. Inteligência artificial. 2. Aprendizagem de máquina. I. Queiroz, Sérgio (orientador). II. Título.

006.3 CDD (23. ed.) UFPE- MEI 2019-018

MARCOS DE SOUZA OLIVEIRA

METODOLOGIA DE SELEÇÃO DE FEATURES NÃO-SUPERVISIONADA PARA CLUSTERING EM CONJUNTO DE DADOS DE ALTA DIMENSIONALIDADE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Aprovada em: 30/08/2018

BANCA EXAMINADORA

_____

Profº. Dr. Sergio Ricardo de Melo Queiroz (Orientador)
Universidade Federal de Pernambuco

_____

Profº. Dr. Francisco de Assis Tenório de Carvalho (Examinador Interno)
Universidade Federal de Pernambuco

_____

Profº. Dr. Renato Fernandes Corrêa (Examinador Externo)
Universidade Federal de Pernambuco

Dedico este trabalho a Deus e a minha família, que foram porto seguro perante as dificuldades durante este percurso.

RESUMO

Em mineração de dados, a seleção de features é uma tarefa importante na eliminação de features irrelevantes/redundantes do conjunto de dados. Na aprendizagem de máquina não-supervisionada, a seleção de features é considerada ainda mais difícil do que na aprendizagem supervisionada, por não possuir a informação de classe, que possa ser utilizada para a avaliação das features. Muitos métodos de seleção de features na aprendizagem não-supervisionada são propostos na literatura, porém a avaliação do melhor conjunto de features é realizada através de critérios supervisionados, onde as classes são exigidas, o que nem sempre ocorre em um cenário real. Outro problema é que os métodos atribuem scores para cada feature e utilizam números mágicos para escolher as m-melhores features. Assim, neste trabalho é proposta uma metodologia que tentará ajudar especialistas de dados a responder questões simples, mas importantes, como: (1) os métodos de seleção de features existentes possuem um resultado similar? (2) Existe um método consistentemente "melhor" ? Geralmente, esses métodos ordenam os atributos baseando-se em um score. Portanto, em relação aos resultados obtidos pelos métodos surgem algumas questões importantes: (3) Se selecionarmos m-melhores features, qual m será considerado o melhor? Além disso, muitos desses métodos não são totalmente livres de parâmetros, nos remetendo a uma outra questão: (4) Como selecionar bons parâmetros para os métodos em um cenário não-supervisionado? Outra questão interessante é: (5) Assumindo que nós temos diferentes opções para os métodos de seleção de features, poderíamos obter melhores resultados se selecionarmos as features usando uma combinação de métodos? Se sim, então como podemos combinar os métodos? Neste trabalho nós analisamos essas questões e propomos uma metodologia que irá realizar a seleção de features não-supervisionada para clustering em conjunto de dados de alta dimensionalidade. Nós avaliamos a metodologia proposta em vários conjuntos de dados de domínios como processamento de imagens e bioinformática. Os resultados mostraram que através do subconjunto de features sugerido pela metodologia é possível obter resultados melhores para os indicadores de acurácia, NMI e Corrected Rand, do que quando utilizado o conjunto original de features. Ao final também elencamos melhorias a serem realizadas em trabalhos futuros com potencial de melhorar o desempenho já obtido.

Palavras-chaves: Feature Selection. Clustering. Aprendizagem não-supervisionada. Redução de dimensionalidade.

ABSTRACT

In Data Mining, feature selection is an important task to eliminate uninformative features from datasets. In unsupervised learning, the selection of features is considered even more difficult than in supervised learning, we do not have any class information, that can be used to evaluate the features. Many feature selection methods in unsupervised learning are proposed in the literature, but the evaluation of the best subset of features is performed through supervised criteria, where class labels are required, which does not always occur in a real scenario. Another problem is that the methods assign scores for each feature and use magic numbers to choose the m-better features. Thus, in this work is proposed a methodology that will try to help data specialists to answer simple but important questions, such as: (1) do the existing features selection methods have a similar result? (2) Is there a consistently "better" method? Generally, these methods rank attributes based on a score. Therefore, in relation to the obtained results by the methods, some important questions arise: (3) If we select m-better features, what m will be considered the best? In addition, many of these methods are not fully parameter-free, referring to another question: (4) How to select good parameters for the methods in an unsupervised scenario? Another interesting question is: (5) Assuming we have different options for feature selection methods, could we get better results if we select features using a combination of methods? If yes, then how can we combine the methods? In this work we analyze these questions and propose a methodology that will perform the unsupervised feature selection for clustering in high dimensional data sets. We have evaluated the methodology proposed in several data sets from bioinformatics and image processing domains. The results showed that by using subsets of features suggested by the methodology it is possible to obtain better results for the indicators of accuracy, NMI and Corrected Rand, that when using the original set of features. However, it seens that there are future improvements to be made with potential to increase the performance already obtained.

Keywords: Feature Selection. Clustering. Unsupervised learning. Dimensionality Reduction.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Com a grande disponibilidade de informações através do avanço da internet, áreas como Big Data e Data Mining tem chamado bastante atenção nos últimos anos, surgindo também inúmeras aplicações em sistemas que utilizam técnicas para a descoberta de padrões a partir dos dados, como atividades de Clustering. Porém também cresce a necessidade do tratamento desses dados, de forma a eliminar informações consideradas irrelevantes/redundantes, sendo inclusive, esta etapa a que consome maior tempo na elaboração de sistemas que realizam o processo de descoberta de conhecimento em bases de dados (KDD) (HAN; PEI; KAMBER, 2011). A Figura 1 mostra as etapas do processo de KDD. Observe que numa das etapas iniciais existe o pré-processamento dos dados, onde podemos realizar, entre outras atividades, a aplicação de algoritmos que eliminam variáveis indesejadas (irrelevante/redundantes) do dataset.

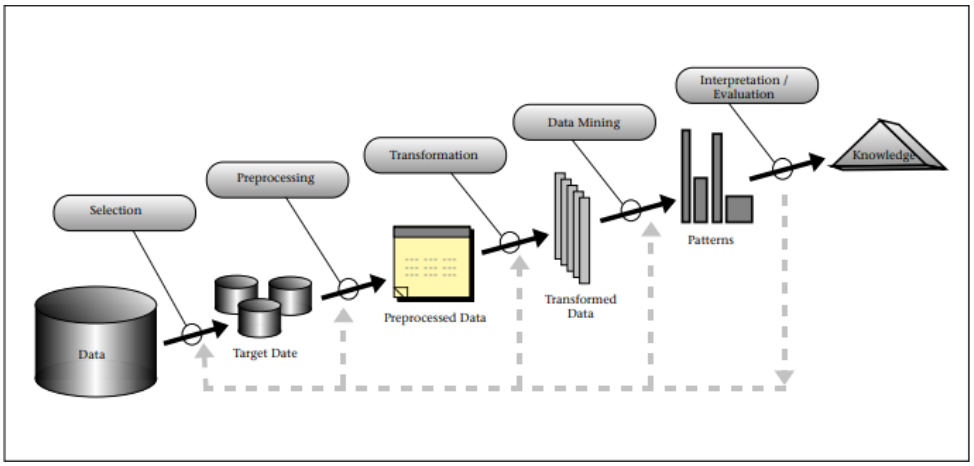


Figura 1 – Processo de Descoberta de Conhecimento em Bases de Dados

Fonte: (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996)

A seleção de Features, características, variáveis ou redução da dimensionalidade como também é chamada, tem se tornado uma importante área de pesquisa nos últimos anos, por causa da necessidade de tratamento dessa grande quantidade de informação disponível. O desafio da seleção de features consiste na eliminação de variáveis consideradas irrelevantes/redundantes em conjuntos de dados. Em alguns domínios, como é visto em bioinformática e em classificação de textos, os objetos, também chamados de instâncias, são descritos por centenas ou até milhares de features. Há muitos benefícios potenciais na seleção de features: facilitar a visualização de dados e sua compreensão, reduzir os requisitos para comparação dos objetos e seu armazenamento, facilitando assim a construção

de modelos, e abordar o problema da maldição da dimensionalidade, melhorando o desempenho de preditores (GUYON; ELISSEEFF, 2003).

1.2 PROBLEMA E MOTIVAÇÃO

Em domínios onde temos uma pequena quantidade de objetos descritos por milhares de features (ex.: Bioinformática), a utilização da seleção de features se torna essencial, nesses cenários nos deparamos com o que é chamado na literatura de "a maldição da dimensionalidade" (DONOHO et al., 2000), onde há uma alta dimensionalidade para uma quantidade pequena de objetos, ou seja, as instâncias são descritas por uma quantidade excessiva de atributos, assim qualquer tentativa de construção de um preditor sem realizar qualquer espécie de seleção de atributos, provocará a obtenção de resultados aleatórios, que podem não representar um padrão, ou não revelar informações relevantes, no mundo real. Desta forma, a seleção de features é um passo essencial no processo de descoberta de conhecimento nesses tipos de conjunto de dados.

Diferentemente da extração, a seleção de Features visa selecionar um subconjunto de Features em um dataset sem realizar quaisquer transformações no conjunto de Features originais, já a extração de Features realizará essas transformações, de forma a obter um novo conjunto de Features. Esta técnica é conhecida como o mapeamento do dataset original para um conjunto de Features em baixa dimensionalidade, a ideia consiste em encontrar o subconjunto de Features que mais se aproxime dos dados originais, geralmente essa aproximação se dá pela minimização do erro quadrado médio ou alguma medida de distorção (LIU; MOTODA, 2007). Um exemplo de algoritmo que realiza este procedimento é o de análise de componentes principais (PCA).

Neste trabalho iremos focar na seleção de features na aprendizagem de máquina não supervisionada, onde não há informação de classes para os objetos, assim não teremos um conjunto de dados de treinamento com a informação dos labels. Tal escolha é devido a vários motivos, um deles é porque o preenchimento da informação do label em datasets com grande quantidade de dados para humanos não é trivial além de ser subjetivo (LIU; MOTODA, 2007). Além da possibilidade de existirem categorias não conhecidas por especialistas presentes nos dados. Por causa desses benefícios, a aprendizagem de máquina não-supervisionada tem atraído muita atenção. Mas é considerada mais difícil do que na aprendizagem supervisionada por não possuir a informação dos labels dos objetos para a construção dos preditores.

Na literatura são propostos alguns métodos para a seleção de features não-supervisionada. Esses métodos, geralmente, realizam um ranqueamento atribuindo uma espécie de score para cada feature. Os especialistas de dados precisam escolher as m features mais bem posicionadas no ranking. Assim é necessário estabelecer uma metodologia para escolher o valor mais apropriado para o m, de forma a se obter um bom ponto de corte. Os trabalhos que tratam a seleção de features não-supervisionada não focam este problema, e executam

o método proposto com “números mágicos“ para o m, como $[10, 15, 20, 25...]$, e avaliam o resultado utilizando um método supervisionado, por meio de um conjunto de dados onde as classes são conhecidas, de forma a verificar quão similar os padrões encontrados (através do método não-supervisionado) utilizando esses pontos de corte são com as partições já conhecidas.

1.3 QUESTÕES E OBJETIVOS DA PESQUISA

Um problema prático é que quando especialistas de dados utilizam seleção de features em um cenário real não-supervisionado, não há classes previamente estabelecidas que possam ser usadas para avaliar quão bom é o m selecionado, assim a escolha do m precisa ser feita de forma não-supervisionada. Outro problema é que os métodos de seleção de features não-supervisionados não são totalmente livres de parâmetros, necessitando desta forma a aplicação de uma metodologia para a escolha do conjunto de parâmetros para o devido método. Por fim, existe também o problema de qual método de seleção de features utilizar, com a disponibilidade de alguns dos métodos, há algum método consistentemente “melhor” para conjuntos de dados de alta dimensionalidade? Existem métodos que possuem resultados similares? É possível obter resultados melhores caso seja feita uma fusão dos resultados individuais desses métodos?

Nós podemos resumir esses problemas de interesse em cinco questões:

(Q1) Há algum método que obtém um resultado similar a outro que possa ser considerado como equivalente?

(Q2) Há algum método considerado ”melhor” do que outros para diferentes conjuntos de dados com alta dimensionalidade?

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

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

(Q5) É possível obter resultados melhores se forem utilizadas Features a partir da junção de diferentes resultados de métodos de seleção? Como podemos juntar os resultados dos métodos de forma eficiente?

Diante dessas questões, este trabalho tem como objetivo principal propor uma metodologia de seleção de Features a ser executada em um cenário totalmente não-supervisionado, tendo como objetivo secundário a busca por alternativas diferentes das que já existem na literatura para tentar responder as questões levantadas.

1.4 METODOLOGIA

Para atender aos objetivos desta pesquisa será realizado um experimento em oito conjuntos de dados, publicamente disponíveis, sendo cinco desses conjuntos na área de Bioinformática e três na área de processamento de imagens. Existem outros domínios que possuem datasets com alta-dimensionalidade, como na classificação de textos, porém este trabalho se limitou a utilizar apenas as áreas de Bioinformática e processamento de imagens para não exaurir todas as possibilidades através dos experimentos. A ideia é utilizar a metodologia em datasets de diferentes domínios e averiguar sua aplicabilidade, de forma a torná-lo genérico para outras possíveis áreas. O ambiente de desenvolvimento utilizado buscará atender as configurações de um computador convencional, facilitando assim a replicação do estudo. Desta forma foram feitas as seguintes atividades:

(M1) Estudo da literatura sobre os conceitos relacionados e os métodos de seleção de Features aplicados em atividades de clustering em um cenário parcial, ou totalmente, não-supervisionado;

(M2) Levantamento dos conjuntos de dados. Nesta etapa foram verificadas, inicialmente, bases de dados na área de Bioinformática, depois utilizou-se conjuntos de dados para reconhecimento facial. A premissa para a escolha desses dados foi a de possuir as classes dos objetos, para poder efetuar um estudo comparativo com índices de avaliação supervisionada e não-supervisionada;

(M3) Planejamento e desenvolvimento da metodologia de seleção de Features, com as etapas:

- Normalização dos dados: busca tornar os valores dos dados em uma mesma escala;

- Geração de vários rankings de Features a partir da execução de vários métodos: cada método possui várias configurações, assim cada configuração resultará em um ranking de Features, a configuração será escolhida através de um critério não-supervisionado, a maior média das silhuetas nos clusters formados (cf. Seção 2.3.2), questão (Q4);

- Junção de métodos: combinação entre os rankings gerados pelos métodos, na tentativa de responder a questão (Q5);

- Seleção das m-melhores Features: utilizar algumas abordagens, além dos números mágicos já utilizados na literatura, para definir o ponto de corte, questão (Q3);

(M4) Avaliação do subconjunto de Features, selecionado pelo maior valor de silhueta, e o que poderia ser considerado o "melhor" resultado, de forma a verificarmos o desenvolvimento de nossa metodologia.

Esta pesquisa seguirá uma metodologia quantitativa, onde buscará através de algumas análises justificar a viabilidade e utilização de uma nova metodologia para a seleção de Features em um cenário totalmente não-supervisionado.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este trabalho está organizado da seguinte forma: o capítulo 2 irá fazer uma revisão na literatura, a fim de abordar os principais fundamentos sobre a atividade de agrupamento em conjuntos de dados. Nele é descrito o algoritmo k-means e, em seguida, critérios de avaliação de agrupamentos obtidos. Também são descritos métodos não-supervisionados de seleção de features e suas principais características, bem como a análise de componentes principais (PCA), que será usada para uma avaliação visual dos resultados. O capítulo 3 irá detalhar a metodologia proposta, assim como o seu funcionamento. No capitulo 4 serão relatados os experimentos realizados, enquanto que o capítulo 5 irá analisar os resultados obtidos pelos experimentos. No capítulo 6 será realizada uma análise final, relatando as conclusões obtidas nesta pesquisa e melhorias que poderão ser exploradas em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo se realizará uma revisão da literatura de forma a fundamentar os principais tópicos relacionados com a metodologia de seleção de Features a ser proposta. Esses tópicos se referem à atividade de agrupamento em conjunto de dados, à seleção de Features e os algoritmos utilizados.

2.1 CLUSTERING

Devido à grande quantidade de dados disponível, áreas como Big Data e Data Mining tem atraído bastante atenção nas últimas décadas, e cresce também a necessidade de categorização dessas informações, auxiliando na identificação e análise de padrões. Outro problema é que o preenchimento manual das categorias para essa grande quantidade de informações é uma atividade bastante difícil para humanos, além de estar sujeito à subjetividade na interpretação de quem esteja preenchendo. Esta atividade manual também impossibilita a descoberta de padrões ainda não conhecidos por especialistas. Desta forma, o preenchimento do label dos dados de forma automática torna-se um passo indispensável para a mineração de dados(ALELYANI; TANG; LIU, 2013).

O agrupamento de dados, Clustering, é uma das mais populares técnicas para a categorização automática dos dados. A atividade de Clustering consiste no agrupamento de objetos considerados similares. Essa similaridade é calculada através de alguma medida de distância entre objetos. As medidas de distância mais utilizadas são a Euclideana e Manhattan (SOLER et al., 2013).

Muitos são os trabalhos que podem ser feitos com a aplicação de algoritmos de Clustering, tais como algoritmos de detecção de objetos a partir da formação de clusters entre os pixels (KRÄHENBÜHL; KOLTUN, 2014). Entre os algoritmos disponíveis na literatura para a atividade de Clustering, neste trabalho será utilizado o k-means, que se trata um dos mais populares.

2.2 K-MEANS

O algoritmo consiste em dividir os objetos em k grupos, clusters, após a utilização de alguma medida de distância, como a distância euclideana. Cada cluster possui um valor central chamado “centroide”, este valor consiste na média entre os objetos pertencentes àquele cluster. Observe que este valor nem sempre representa um objeto existente. A versão do algoritmo de k-médias utilizado neste trabalho está disponível em (LI et al., 2016). O método busca minimizar a soma total das distâncias intra-cluster, *Total Sum*

Intra-Cluster Distance (TSICD), também chamada de inércia, definida por:

$$\text{TSICD} = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{f=1}^{F} (x_{if} - \bar{x}_{kf})^2 \tag{2.1}$$

Onde K é o número de clusters, i é um objeto do cluster C_k com F Features, x_{if} é o valor para a Feature f do objeto i e $\bar{x}_{kf}$ é o valor para a Feature f para o centroide do cluster k.

A TSICD busca maximizar a coerência interna dos clusters (ARTHUR; VASSILVITSKII, 2007). Assim, quanto menor for o valor da inércia melhor será considerado o cluster. Porém existem alguns potenciais problemas no uso desta medida:

1. A inércia considera que os clusters são convexos e isotrópicos, o que nem sempre é verdade;

2. Inércia não é uma medida normalizada, sabemos que quanto menor o seu valor melhor será considerado o cluster e zero seria um cluster ótimo. Porém quanto maior for o número de Features, tipicamente maior será a TSICD, o que, por exemplo, inviabiliza o uso dessa métrica para comparação de clusters gerados a partir de conjuntos de features diferentes para os mesmos objetos.

O k-means é um algoritmo iterativo, que pode ser descrito pelo pseudo-código do Algoritmo 1. Inicialmente são (aleatoriamente) escolhidos os k centroides iniciais. Em seguida realiza-se um loop onde inicialmente cada objeto é atribuído ao grupo do centroide mais próximo, então são recalculados os centroides de cada grupo de maneira a minimizar a distância intra-cluster de cada um deles. O procedimento se repete até que não haja alterações nos clusters, ou um número máximo de iterações seja alcançado.

Algoritmo 1 Algoritmo do K-Means

```
1: procedure K-MEANS(k: clusters, D: conjunto de dados)
2:     escolher aleatoriamente k objetos de D como centroides dos clusters iniciais;
3:     repeat
4:         (re) atribuir cada objeto ao cluster do centroide mais próximo;
5:         atualizar o centroide de cada cluster de forma a minimizar a distância intra-
   cluster;
6:     until clusters sem alterações;
7:     return conjunto de k clusters;
8: end procedure
```

Como é possível observar no pseudo-código acima, é preciso informar o número de clusters para a execução do k-means. Existem várias abordagens na literatura para definir este valor, porém nos experimentos deste trabalho será utilizado o número de classes pré-existentes, conhecido em todos os conjuntos de dados utilizados, como sendo o número de clusters. Trabalhos futuros poderão tratar da descoberta deste valor, como é realizado em (TIBSHIRANI; WALTHER; HASTIE, 2001).

2.3 MEDIDAS DE AVALIAÇÃO EM CLUSTERING

Um dos desafios na aprendizagem não-supervisionada é avaliar quão bom é um determinado método, devido à ausência da categorização dos dados. Em Clustering existem três tipos de medidas numéricas utilizadas para avaliar a qualidade dos clusters: índices internos, externos e relativos (RENDÓN et al., 2011). Índices do tipo externo comparam os clusters formados com uma estrutura conhecida previamente, ou seja, é necessário um conhecimento prévio sobre os dados para a comparação entre as estruturas previstas e geradas. Índices do tipo interno buscam avaliar informações intrínsecas aos clusters, sem fazer uso de quaisquer informações prévias. Já os índices relativos são usados para comparar diferentes soluções de clustering em relação a índices externos ou internos. Este trabalho buscará seguir uma metodologia não-supervisionada, assim os critérios do tipo interno serão utilizados pela metodologia proposta. No entanto, índices externos de informação mútua normalizada (NMI, do inglês, *normalized mutual information*) e *Corrected Rand* serão utilizados para avaliarmos os resultados.

2.3.1 AVALIAÇÃO SUPERVISIONADA

Entre os índices supervisionados para avaliação da performance de clustering, utilizaremos neste trabalho o *normalized mutual information* (NMI), o *Corrected Rand* e a *acurácia*.

2.3.1.1 NMI

O NMI é um importante índice utilizado em muitos trabalhos presentes na literatura, tais como (YAO et al., 2015), o índice é uma normalização do score de informação mútua entre as partições geradas pelo algoritmo de clustering e as classes dos objetos, onde o propósito é retornar um resultado entre 0 e 1, sendo 0 quando não há nenhuma informação mútua entre os conjuntos e 1 quando há uma perfeita correlação (STREHL; GHOSH, 2002). A definição do cálculo para o NMI é feito da seguinte forma: considere dois conjuntos, o conjunto predito (U) e o conjunto previsto (V), calculamos a entropia (H) para ambos os conjuntos:

$$\mathrm{H(U)} = -\sum_{i=1}^{|U|} P(i)\log(P(i)) \tag{2.2}$$

$$\mathrm{H(V)} = -\sum_{j=1}^{|V|} P(j)\log(P(j)) \tag{2.3}$$

Onde P(i) é a probabilidade de um objeto aleatório de U ser atribuído à classe Ui, da mesma forma para P(j). Após calcularmos as entropias calculamos o índice de informação

mútua (MI) para os conjuntos U e V:

$$\mathrm{MI(U,V)} = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} P(i,j)\log(\frac{P(i,j)}{P(i)P'(j)}) \tag{2.4}$$

Onde $P(i,j) = |U \cap V|/N$, e N sendo o total de objetos. O MI também pode ser escrito da seguinte forma:

$$\mathrm{MI(U,V)} = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N}\log(\frac{N|U_i \cap V_j|}{|U_i||V_j|}) \tag{2.5}$$

O índice de informação mútua normalizado, *normalized mutual information* (NMI), pode ser definido como:

$$\mathrm{NMI(U,V)} = \frac{MI(U,V)}{média(H(U),H(V))} \tag{2.6}$$

2.3.1.2 CORRECTED RAND

O índice de comparação entre partições *Corrected Rand*, ou também conhecido na literatura como *adjusted Rand*, se trata de um método para comparar dois conjuntos (neste trabalho consideraremos a comparação entre as categorias dos objetos com os grupos gerados pelo algoritmo de cluster)(HUBERT; ARABIE, 1985). Com esta métrica é possível obter o resultado de forma simétrica, ou seja, não haverá diferença ao se comparar o conjunto predito com o conjunto categorizado previamente ou vice-versa, também é possível obter resultados negativos ou próximos de zero quando os dois conjuntos comparados sejam bastante diferentes, ou seja, o resultado deste índice varia de -1 a 1.

Considere C como o conjunto das classes dos objetos e K o conjunto de clusters gerados pelo k-means, assim temos:

1. a, o número de pares de elementos que coincidem nas listas C e K;
2. b, o número de pares de elementos que não coincidem nas listas C e K;

O *Rand Index* (RI) é definido da seguinte forma:

$$\mathrm{RI} = \frac{a+b}{\binom{N}{2}} \tag{2.7}$$

Onde $\binom{N}{2}$ se refere ao total de pares, não ordenados, de N elementos. Por exemplo, suponha que tenhamos uma lista de seis objetos (a, b, c, d, e, f), N = 6, assim temos 15 possíveis pares: {a, b}, {a, c}, {a, d}, {a, e}, {a, f}, {b, c}, {b, d}, {b, e}, {b, f}, {c, d}, {c, e}, {c, f}, {d, e}, {d, f}, e {e, f}. Considerando que as categorias dos objetos sejam representadas por $C = \{1,1,2,2,3,3\}$, e que os clusters atribuídos para os objetos

sejam $K = \{1, 1, 1, 2, 2, 3\}$, teremos $a = 1$, pois apenas o par {a, b} possui as mesmas atribuições em ambos os conjuntos C e K, e $b = 8$, pois os pares {a, d}, {a, e}, {a, f}, {b, d}, {b, e}, {b, f}, {c, e} e {c, f} não coincidem com as mesmas atribuições nos conjuntos C e K, assim o $RI = \frac{1+8}{15} = 0,6$.

O RI não considera as atribuições feitas aos objetos de forma aleatória, ou seja, não é garantido que o valor para o RI seja próximo de zero quando informamos o número de clusters igual ao número de objetos, para resolver este problema é realizada uma penalização através do E(RI), que se refere ao índice de RI esperado. Desta forma, para calcularmos o *Corrected Rand* (CR) utilizamos a seguinte equação:

$$\text{CR} = \frac{RI - E(RI)}{\max(RI) - E(RI)} \tag{2.8}$$

2.3.2 AVALIAÇÃO NÃO-SUPERVISIONADA

A avaliação interna dos clusters utiliza dois tipos de avaliação, intra e inter clusters. A avaliação intra-cluster busca avaliar a densidade do agrupamento, ou seja, quão próximo estão os objetos do cluster para com o seu centroide, este conceito é também conhecido como homogeneidade interna. A avaliação inter-cluster avalia a distância entre os clusters, o quão separado estão os agrupamentos (heterogeneidade). A Figura 2 demonstra essa diferença entre as avaliações inter e intra cluster.

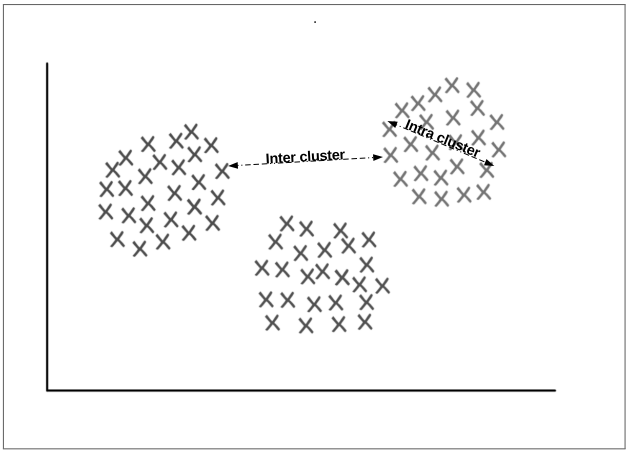


Figura 2 – Inter e Intra Cluster

A medida utilizada neste trabalho, como já mencionada anteriormente, será baseada na silhueta, mais especificamente, a média das silhuetas de todos os objetos. Dessa forma, buscaremos após as iterações do k-means obter agrupamentos mais coesos (homogeneidade) e separados (heterogeneidade). 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:

$$\text{s(i)} = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2.9}$$

O valor da silhueta pode estar no seguinte intervalo: $-1 \leq s(i) \leq 1$ (quanto maior, melhor), onde:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & se \quad a(i) < b(i) \\ 0, & se \quad a(i) = b(i) \\ b(i)/a(i) - 1, & se \quad a(i) > b(i) \end{cases} \tag{2.10}$$

2.3.3 ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)

A Análise de Componentes Principais (ACP) ou Principal Component Analysis (PCA) é uma importante técnica de análise estatística utilizada em muitas áreas. O objetivo da análise consiste em extrair informações importantes de um conjunto de dados e expressar essas informações na forma de vetores ortogonais chamados de componentes principais (ABDI; WILLIAMS, 2010). Essa transformação nos dados possibilita observar, através dos componentes principais, a variabilidade dos dados. Cada componente representa uma dimensão nos dados. Onde o primeiro componente corresponde a dimensão de maior variância, o segundo o de segunda maior variância, respectivamente. A Figura 3 mostra um exemplo onde através dos dois primeiros componentes é possível cobrir boa parte da variabilidade dos dados.

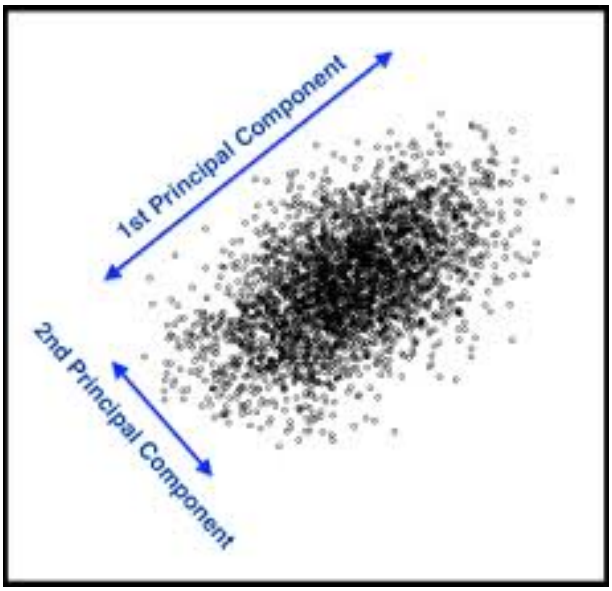


Figura 3 – Principais Componentes

Nos experimentos realizados neste trabalho será utilizada a análise de componentes principais, de forma a verificar possíveis padrões encontrados pelas Features obtidas na execução da metodologia e também servir como insumo para validar a performance obtida. Assim, de forma visual, poderemos avaliar a formação clara, ou não, de clusters através de duas dimensões (os dois primeiros componentes), que geralmente cobrem a maior variabilidade nos dados.

2.4 MÉTODOS FILTER, WRAPPER E EMBEDDED

Existem três categorias de métodos para a seleção de Features[1]: filter, wrapper e embedded (LIU; MOTODA, 2007). Métodos do tipo *filter* realizam a seleção sem a necessidade do treinamento de algoritmos de aprendizagem, é utilizado algum tipo de propriedade estatística nos dados, como uma matriz de Gauss e Laplace (HE; CAI; NIYOGI, 2006), esses métodos tendem a selecionar features redundantes por não avaliar correlações no subconjunto de features selecionadas. Métodos filter também tendem a serem mais rápidos.

Métodos do tipo *wrapper* utilizam algum tipo de algoritmo de aprendizagem para escolher as features, esses algoritmos costumam ter maior complexidade computacional do que os métodos do tipo filter. O procedimento consiste em classificar "boas" features 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 essas Features resultantes são avaliadas pelo algoritmo de aprendizagem. Dos métodos utilizados neste trabalho, os métodos SPEC (ZHAO; LIU, 2007) e Laplacian Score (HE; CAI; NIYOGI, 2006) são do tipo *filter*. O método iDetect (YAO et al., 2015) é do tipo *wrapper* e o método GLSPFS (LIU et al., 2014) é do tipo *embedded*.

Alguns outros métodos de seleção de Features não-supervisionada são propostos na literatura, tais como MCFS (CAI; ZHANG; HE, 2010), UDFS (YANG et al., 2011), FSASL (DU; SHEN, 2015), JELSR (HOU et al., 2011) e Sparse K-Means (WITTEN; TIBSHIRANI, 2010). Porém, para a execução desses métodos foi necessário um poder de processamento computacional elevado, quando aplicados para a seleção de features em conjunto de dados de alta dimensionalidade, impossibilitando a utilização dos mesmos em um computador convencional. Acreditamos, em um trabalho futuro, no desenvolvimento de melhorias nas implementações desses métodos, tais como propor uma execução paralela através do processamento de uma unidade gráfica como a GPU. A seguir serão explicados, de forma resumida, os conceitos e os parâmetros de funcionamento dos métodos de seleção de features que serão utilizados neste trabalho.

[1] Um campo relacionado é a extração de features, onde é gerado um conjunto de novas Features a partir de um conjunto original de dados (Ex.: análise de componentes principais –PCA). No entanto, iremos focar na seleção de Features, que selecionará as Features mais relevantes a partir de um conjunto de Features.

2.4.1 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 (YAO et al., 2015). A Figura 4 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.

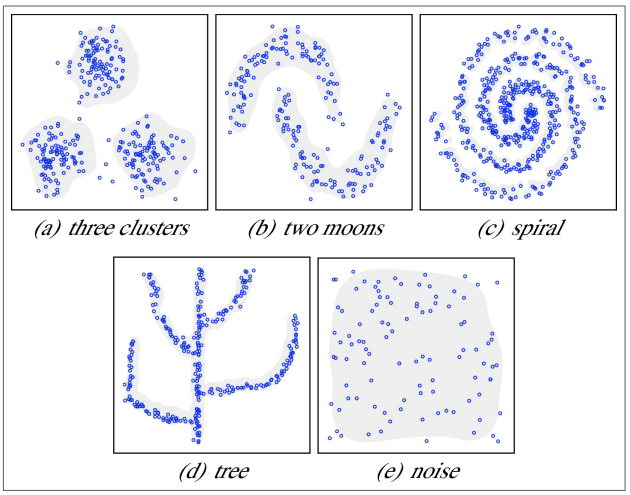


Figura 4 – Exemplos de Possíveis Estruturas Presentes em Conjunto de Dados

Fonte: (YAO et al., 2015)

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 4, 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 foi relaxado para que as variáveis pudessem ser reais, utilizando o conceito de entropia negativa (FRIEDMAN; MEULMAN, 2004), 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 features 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 (TIBSHIRANI, 1996), 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 (TIBSHIRANI; WALTHER; HASTIE, 2001).

Nos experimentos realizados no trabalho que define o iDetect (YAO et al., 2015), observou-se que sua execução se deu em alguns segundos, mesmo para datasets onde o número de features 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.

2.4.2 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 (HE; CAI; NIYOGI, 2006). 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 das features é de acordo com sua preservação em uma estrutura local, também denominado como Locality Preserving Projections (LPP) (HE; NIYOGI, 2004).

Considere que tenhamos um matriz $n \times m$, onde n se refere aos objetos e m as features. 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:

$$\mathrm{L} = D - W \tag{2.11}$$

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 uma feature f da seguinte forma:

$$\mathrm{LS(f)} = \frac{\tilde{f}^T L \tilde{f}}{\tilde{f}^T D \tilde{f}}, onde \tilde{f} = f - \frac{f^T D 1}{I^T D 1} 1, \tag{2.12}$$

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.

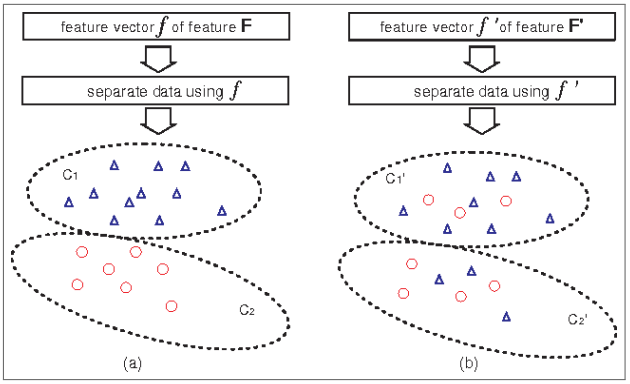
O parâmetro weight mode possui duas opções, binário e o heat kernel, para calcular o peso que será atribuído sobre as arestas. O modo binário, o modo mais simples, atribui 1 como peso para as arestas dos objetos que estiverem conectados. O heat kernel, também conhecido como RBF kernel, trata-se de uma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

$$\mathrm{S_{ij}} = e^{-\frac{||x_1 - x_2||^2}{(2)}} \tag{2.13}$$

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

2.4.3 SPEC

O SPEC propõe uma solução de seleção de features para atender situações de aprendizado supervisionado e também não-supervisionado. Ele é baseado na teoria espectral de grafos (CVETKOVIĆ; DOOB; SACHS, 1980). 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 as features que favoreçam para a formação de estruturas mais consistentes.



Fonte: (ZHAO; LIU, 2007)

Figura 5 – Clusters Formados pelo Espectro de Grafos

A Figura 5 mostra os possíveis Clusters formados a partir das informações de estruturas obtidas pelo espectro do grafo. Onde é possível observar que a Feature F é mais relevante para a definição de clusters do que a Feature F'.

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:

$$\mathrm{pG}(\lambda) = det(\lambda I - M) \tag{2.14}$$

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(λ1),..., 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 (ABREU et al., 2007).

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 feature F_1 é utilizada a seguinte equação:

$$\mathrm{Score(F_1)} = \frac{f^T L f}{f^T D f}, (LaplacianScore) \tag{2.15}$$

   Onde f se refere ao vetor de valores da feature 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:

$$\mathrm{Score(F_1)} = \frac{\sum_{j=1}^{n-1} \alpha_j^2 \lambda_j}{\sum_{j=1}^{n-1} \alpha_j^2} \tag{2.16}$$

   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 das features nesta opção é definido a seguir:

$$\mathrm{Score(F_1)} = \sum_{j=1}^{n-1} (2 - \lambda_j)\alpha_j^2 \tag{2.17}$$

Observe através do parâmetro *style*, que quando utilizamos a primeira opção (-1), o score atribuído para as features 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 iremos utilizar essa opção.

O método SPEC segue a teoria que uma feature 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 features 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 feature.

2.4.4 GLSPFS

O GLSPFS (LIU et al., 2014) trata-se de um método capaz de realizar a seleção de features em um cenário supervisionado, não-supervisionado e também semi-supervisionado (seleciona as features 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 (GU et al., 2011), 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) (ROWEIS; SAUL, 2000), locality preserve projection (LPP) (HE; NIYOGI, 2004) e local tangent space alignment (LTSA) (ZHANG et al., 2007). Neste trabalho iremos utilizar esses três algoritmos. 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.

3 METODOLOGIA MUI

Neste trabalho será desenvolvida uma metodologia capaz de efetuar a seleção de Features em um cenário não supervisionado, considera-se que o subconjunto de Features resultante seja capaz de gerar modelos mais consistentes, auxiliando especialistas de dados a obterem informações relevantes presentes nos dados. Essas informações serão obtidas a partir da formação de agrupamentos nos conjuntos de dados, utilizaremos a execução de um k-means clássico, conforme abordado no capítulo anterior. Porém, antes de descrever o fluxo de funcionamento da metodologia, é importante definir algumas restrições que serão seguidas:

(1) A metodologia precisa ser independente de métodos de seleção de Features, ou seja, deve ser possível a fácil inclusão/remoção dos algoritmos para pontuar as Features, 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 Features 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 de Features, possibilitando, inclusive, a junção entre eles, com o objetivo de obter os melhores resultados. Este requisito possibilita diversificar a seleção das Features, pois o resultado de um método pode ser complementar a outro. [MÚLTIPLO].

Para melhor referenciar as restrições listadas acima, foram incluídas as palavras-chaves, entre colchetes, ao final de cada item, assim a restrição [INDEPENDENTE] será atendida ao assumir que cada método de seleção de Features a ser utilizado, realize alguma espécie de ordenação ou pontuação para as Features, assim um novo método poderá ser adicionado sem ter que alterar as etapas de funcionamento da metodologia.

Para atender a restrição [NÃO-SUPERVISIONADO], será feito uso de um algoritmo de Clustering com base na otimização de um critério, como a minimização do total da soma dos quadrados intra-cluster no k-means, após a execução do algoritmo será calculada a silhueta para cada agrupamento, em seguida utilizaremos a média das silhuetas como critério de escolha sobre os subconjuntos de Features utilizados, assim quanto maior for

este valor melhor será considerado as Features selecionadas. Na seção 3.5 iremos explicar como será feito este procedimento na execução do k-means, porém em um trabalho futuro outro critério possa ser adicionado, contanto que o mesmo não tenha a exigência da informação do label dos objetos.

Por fim, para a restrição [MÚLTIPLO], a seção 3.3 mostra como serão utilizadas múltiplas configurações em cada método de seleção de Features e também a possível combinação dos resultados para cada método. Ao final, será disponibilizado ao especialista de dados uma lista dos melhores subconjuntos de Features e qual seria o subconjunto escolhido pela nossa metodologia, assim caberia ao mesmo escolher quais Features seriam consideradas de fato as melhores.

Para facilitar a referência sobre a metodologia proposta neste trabalho, utilizaremos as iniciais em inglês de cada restrição e a chamaremos de MUI (*Multiple*, *Unsupervised* e *Independent*). A metodologia MUI pode ser definida como um método de seleção de Features *meta-embedded*, que pode combinar múltiplos tipos de métodos, wrapper e filter, com um algoritmo de Clustering baseado em um critério de otimização, com o objetivo de tentar obter o melhor subconjunto de Features. A metodologia MUI também pode ser definida como um método *ensemble*, capaz de otimizar os parâmetros dos métodos de seleção de Features, para se obter ao final um melhor desempenho. O fluxo de funcionamento da metodologia será mostrado nas seções seguintes, onde pode ser dividido em cinco etapas: normalização, seleção prévia, geração de múltiplos rankings, geração dos m-melhores subconjuntos e a avaliação não-supervisionada.

3.1 NORMALIZAÇÃO DOS DADOS

Em muitos conjuntos de dados é necessário realizar alguma espécie de normalização nos dados para que os mesmos possam ser comparáveis, ou seja, é importante que os dados estejam em uma mesma escala. Este passo é utilizada na literatura para auxiliar classificadores a obterem boas performances, sendo uma importante etapa de pré-processamento de dados em data mining (SHALABI; SHAABAN; KASASBEH, 2006).

Considerando como entrada de dados uma matriz de n (objetos) $\times$ f (Features originais), após a normalização a saída será um novo conjunto de dados $n \times f$, onde cada coluna (Feature) será normalizada para que os valores estejam em uma mesma escala, por exemplo entre 0 e 1. Na seção 4.1 (página 38) será abordado com mais detalhes qual o procedimento adotado para a normalização dos dados utilizado neste trabalho.

3.2 SELEÇÃO PRÉVIA DE FEATURES

Em alguns conjuntos de dados, podemos encontrar features com valores que podem ter pouca variabilidade entre os objetos, isso pode ocasionar um problema na seleção de features que utiliza a qualidade de clusters como critério, os clusters gerados podem

aparentar “boa qualidade” quando observada as distâncias intra-cluster, porém além da homogeneidade é desejável obter também máxima separação entre os clusters, desta forma as features com pouca variabilidade podem dificultar a separação entre os objetos.

Muitos trabalhos que exploram a variabilidade dos dados são propostos na literatura, tais como (SMYTH et al., 2005) e (TUSHER; TIBSHIRANI; CHU, 2001). Uma das mais simples formas de eliminar features de baixa variabilidade é utilizar um filtro de variância. Ou seja, dado que tenhamos uma matriz de entrada $n \times f$, que correspondente à saída da etapa de normalização, após o filtro de variância teremos uma matriz $n \times f_1$, com $f_1 \leq f$, com algumas features (colunas) removidas devido às suas baixas variâncias. Neste trabalho conduzimos um estudo, na seção 4.2 (página 38), para verificar se a utilização de um filtro de variância para eliminar features com baixa variabilidade antes da seleção, utilizando diferentes limiares, implica numa melhor qualidade dos resultados, ou se tal etapa de pré-tratamento dos dados não é necessária dado que serão utilizados métodos de seleção de features em seguida.

3.3 GERAÇÃO DE MÚLTIPLOS RANKINGS

Como foi citado anteriormente, os métodos utilizados neste trabalho necessitam de parâmetros de entrada, assim esta etapa utilizará alguns valores para os parâmetros de cada método de forma a obter melhores resultados, alguns desses parâmetros são críticos, e sua configuração influencia diretamente na obtenção de melhores resultados. Ao invés de testar todas as possibilidades de configuração para os parâmetros, onde isto poderá aumentar o custo computacional, ou utilizar alguma estratégia complexa de otimização, iremos tentar as configurações mencionadas na literatura. Por exemplo, os parâmetros escolhidos para a execução do método GLSPFS foram selecionados com base nos trabalhos de (DU; SHEN, 2015). A Tabela 1 mostra as configurações a serem utilizadas.

Assim, cada configuração de um determinado método resultará em um ranking de Features, onde cada Feature terá um score atribuído pelo método. Como foi pontuado em (Q5) na introdução deste trabalho, iremos tentar realizar a junção dos resultados de diferentes métodos para verificar se suas combinações podem melhorar a performance ao invés de apenas utilizar os resultados dos métodos obtidos após uma execução de forma isolada. Como cada método gera um ranking de Features, iremos escolher a contagem de Borda(BORDA, 1781) para realizar a fusão entre os resultados de cada método. Serão realizadas todas as combinações possíveis entre os quatro métodos (LS, SPEC, iDetect e GLSPFS).

3.3.1 JUNÇÃO DOS RESULTADOS PELA CONTAGEM DE BORDA

A contagem de Borda se trata de um mecanismo de votação utilizado para gerar um ranking global de candidatos (neste trabalho, Features) a partir de rankings individuais

Tabela 1 – Total de Configurações por Método

Método	Parâmetros	Valores	Configurações
LS	matriz W	knn-size = [5,6,7,8]	8
		weight mode = ['binary', heat kernel]	
SPEC	matriz W	knn-size = [5,6,7,8]	16
		weight mode = ['binary', heat kernel]	
	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	

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, feature-b 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)]. Assim, para limitar o número de possíveis combinações será realizada a junção entre os melhores resultados dos métodos. Estes resultados serão obtidos após a avaliação de todas as configurações dos métodos, será considerada a melhor configuração do método a que obter maior valor da silhueta (valor obtido ao utilizar o k-means com o subconjunto de features do ranking gerado pela referida configuração), assim cada método com sua melhor configuração será utilizado na contagem de Borda. A junção será feita com todas as possíveis combinações entre os métodos, combinando os resultados entre dois ou mais métodos. A combinação dos métodos possibilita explorar os dados de diferentes formas (ex.: a metodologia do iDetect é diferente do SPEC), e eles podem se complementar.

Portanto, esta etapa terá como entrada a saída obtida na etapa anterior (uma matriz de dados $n \times f_1$) e como saída os rankings, obtidos pela execução isolada de cada método, considerando as melhores configurações, e os resultados combinados pela contagem de Borda. A Figura 6 mostra o fluxo realizado pela junção dos resultados dos métodos de

seleção de features.

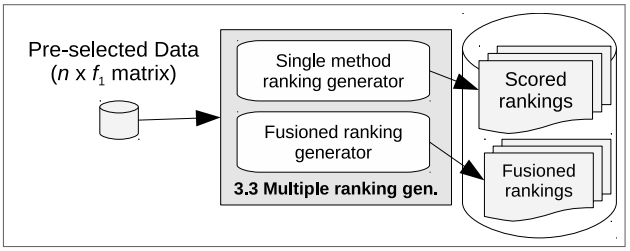


Figura 6 – Processo de Fusão de Métodos

3.4 GERAÇÃO DAS M-MELHORES FEATURES

O problema encontrado nesta etapa se trata da seleção do subconjunto de Features. De acordo com a etapa anterior temos, para cada configuração, um ranking (preenchido com os Scores) de Features. Desta forma, como podemos utilizar tal informação para selecionar o subconjunto de Features? Este subconjunto será escolhido com as m-melhores Features. Assim, o problema é reduzido em encontrar um bom m. Portanto, iremos utilizar três estratégias para a escolha do m: *números mágicos*, *quartil dos Scores* e *ponto de inflexão dos Scores*.

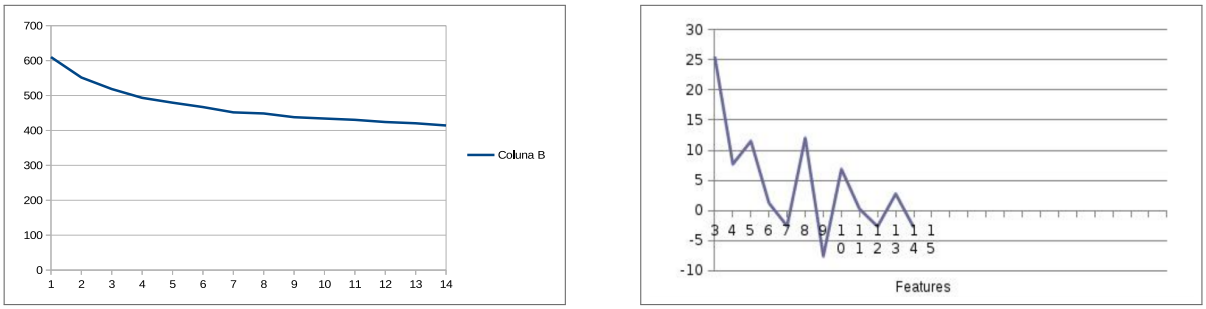
O uso de números mágicos é a estratégia mais simples e também mais frequentemente utilizada na literatura por métodos de seleção de Features, que consiste da simples geração de uma sequência de números para o m, geralmente em uma progressão linear. Por exemplo, inicia com o valor 10 e é incrementado por 5 até um limite de 100, assim obtemos a sequência: $[10, 15, 20, \ldots, 100]$.

Além da utilização de números mágicos, são propostas duas abordagens para a escolha do m. A primeira, quartil dos Scores, seleciona as primeiras m Features de acordo com o quartil, da seguinte maneira: suponha que tenhamos as Features {a, b, c}. Sendo que um método de seleção de Features gerou o seguinte ranking das features, ordenadas segundo seus Scores computados: $\{\mathrm{b} = 0,8, \mathrm{c} = 0,7, \mathrm{a} = 0,2\}$. Se escolhermos o primeiro quartil (25%), manteremos as m features tal que a soma dos seus scores ultrapasse 25% da soma total dos scores. No exemplo, temos que a soma total dos scores é 1.7. A feature b, com $0,8$, representa sozinha 47% da soma total dos scores. Assim, com a escolha do primeiro quartil, manteríamos apenas a feature b. Se escolhêssemos o segundo quartil (50%), teríamos que incluir a feature c. Note que b e c juntas somam um score de $1,5$, que corresponde a 88% do total. Assim se escolhêssemos o terceiro quartil (75%) também manteríamos apenas as features b e c. Observe que, na prática, temos milhares de Features, e precisaremos apenas que as Features não tenham valores negativos para utilizar este método. Se os valores forem negativos, ainda poderemos utilizar este método se for realizada antes a normalização dos valores para torná-los positivos.

A segunda abordagem utiliza o ponto de inflexão dos Scores, este método considera que, conforme o resultado obtido por uma configuração, podemos ordenar as Features de acordo com os Scores de forma descendente e traçar uma linha para representar esses Scores. Foi observado constantemente que esta linha tem pontos de inflexão que podem ser bons candidatos para o m, a partir destes pontos os Scores decrescem mais lentamente do que antes. 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 Features correspondente com a maior segunda derivada. A Figura 7 simula um exemplo de Scores para 15 Features, onde a Feature 8 seria escolhida como ponto de corte, retornando através dessa estratégia as oito primeiras Features. Faremos isto utilizando a seguinte expressão:

$$\max_{f=2}^{F-1} s(f+1) + s(f-1) - 2 * s(f) \tag{3.1}$$

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



(a) Scores em Ordem Descendente (b) Segunda Derivada

Figura 7 – Simulação de Scores para as Features

Este passo tem como sua entrada a saída da etapa anterior, ex.: uma coleção de rankings de Features (uma para cada configuração selecionada), e retorna, para cada ranking recebido como entrada, uma coleção de subconjuntos de Features gerados pelas três diferentes estratégias. A Figura 8 mostra o fluxo de seleção do ponto de corte, a entrada e sua devida saída.

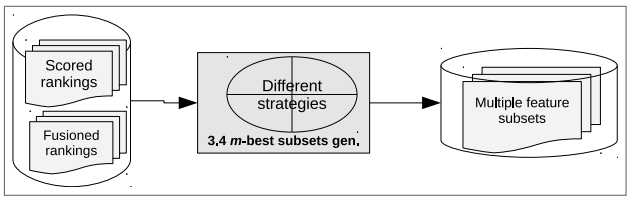


Figura 8 – Seleção das m-melhores Features

Para os rankings obtidos da junção dos resultados pela contagem de Borda, iremos utilizar apenas a abordagem dos números mágicos, pois a abordagem da inflexão e dos

quartis necessita que cada feature tenha um score atribuído, enquanto que o ranking fusionado pela contagem de Borda não possui scores. No capítulo dos experimentos foi realizada uma pesquisa para a seleção do m quando os métodos são executados de forma isolada (as features possuem scores), a ideia é verificar se as abordagens sugeridas (quartil e inflexão) são suficientemente "boas" para substituir a escolha exaustiva do m pelos números mágicos.

3.5 AVALIAÇÃO NÃO-SUPERVISIONADA

Nesta etapa teremos a coleção de subconjuntos de features gerados no passo anterior, em seguida será executado um algoritmo de Clustering para um determinado k em cada subconjunto e avaliado qual obtém melhor performance. Como mencionado no capítulo anterior, existem critérios que validam os Clustering com base nas categorias dos objetos e outros que não dependem dessa informação (RENDÓN et al., 2011): *validação externa* compara os resultados dos clusters com uma partição de referência (os objetos foram categorizados previamente), *avaliação interna* compara os resultados dos Clusters gerados apenas, sem o uso de qualquer referência externa. Assim, como neste trabalho propomos uma metodologia a ser executada em um contexto não-supervisionado, será utilizada a avaliação interna dos Clusters apenas.

Neste trabalho é utilizado o algoritmo clássico do k-means, que possui uma avaliação interna de qualidade como seu próprio critério de otimização, a soma total das distâncias intra-cluster, *Total Sum Intra-Cluster Distance* (TSICD), como é mencionado no capítulo de fundamentação teórica. No entanto, esse critério não pode ser utilizado para avaliar a qualidade dos clusters obtidos com diferentes números de features, uma vez que quanto menos features, menor será a TSICD e vice-versa. Assim, adotaremos a silhueta como base do nosso critério não-supervisionado utilizado para avaliar os clusters. Como é calculada a silhueta para cada objeto, utilizamos como critério a média das silhuetas para selecionar o melhor subconjunto de features. Note que a silhueta, ao contrário da TSICD, é robusta em relação ao número de features, pois é feita uma normalização ao dividir-se pelo valor $\max(a(i), b(i))$ (c.f. seção 2.3.2, página 21)

Ao término desta etapa, teremos, para cada subconjunto de features a média das silhuetas, assim o subconjunto que obter a maior média será armazenado. Lembrando que, como mencionado na seção 3.3.1, obtemos as melhores configurações de cada método de seleção de features isoladamente, em seguida é feita a combinação dos resultados dos métodos, gerando um novo ranking, submetendo-o para a etapa 3.4, com o objetivo de encontrar o melhor ponto de corte para as coleções de features. Desta forma, podemos comparar todas as configurações selecionadas (resultados individuais e da combinação de resultados dos métodos) através das médias das silhuetas que foram armazenadas.

A Figura 9 mostra de forma detalhada o fluxo seguido pela metodologia proposta (MUI), desta forma espera-se que as melhores configurações de cada método revelem

padrões relevantes presentes nos dados, e através do critério não-supervisionado (média das silhuetas) a configuração selecionada possa revelar aquela que seria a escolha por um especialista de dados, tornando a execução do MUI totalmente automatizada, sem a necessidade de uma intervenção humana. Também é importante salientar que espera-se que a metodologia MUI possa realizar um corte considerável no número total de features, reduzindo assim a dimensionalidade (complexidade) do modelo anteriormente utilizado. Em seguida iremos avaliar se essas premissas serão atendidas através do capítulo experimentos.

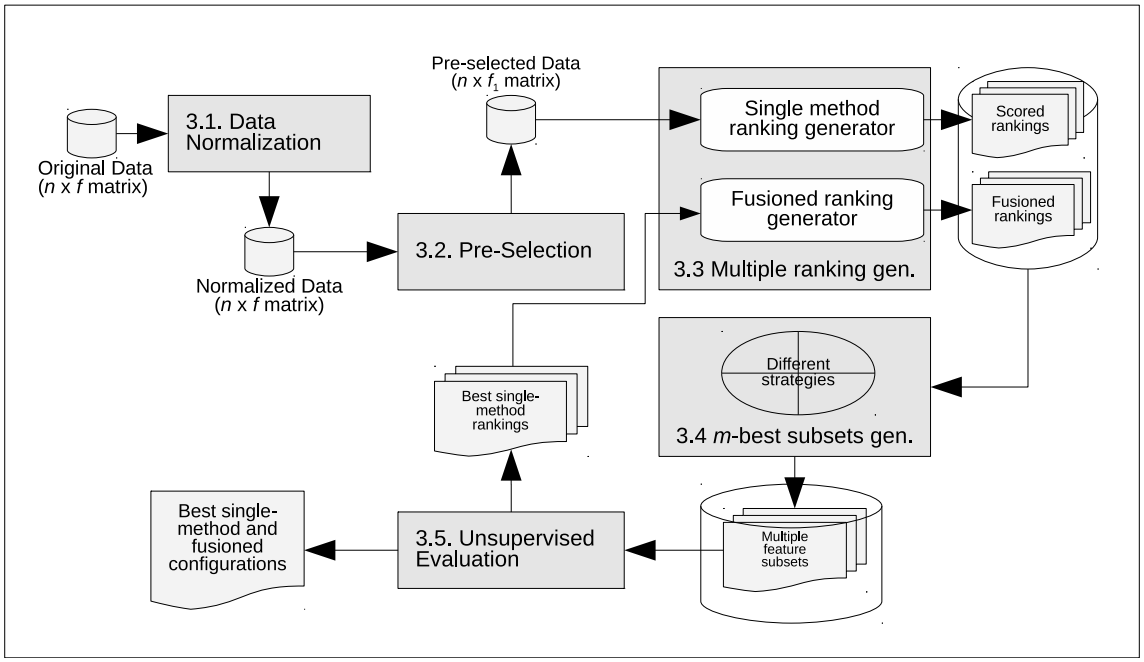


Figura 9 – Workflow do MUI

Como podemos observar, todas as etapas dependem que os scores das features, atribuídos pelos métodos de seleção propostos na literatura, sejam coerentes (as features mais relevantes obtenham melhores pontuações), pois mesmo que o MUI busque otimizar a escolha daquele que seria considerado o melhor subconjunto de features, apenas um bom ranqueamento das features possibilitará a obtenção de sucesso pela metodologia. Nos capítulos seguintes os experimentos realizados na metodologia MUI, bem como seus devidos resultados, serão apresentados.

4 EXPERIMENTOS

Neste capítulo iremos executar a metodologia proposta (MUI) em oito diferentes conjuntos de dados de alta dimensionalidade do mundo real, esses datasets estão publicamente disponíveis, onde é possível obter os dados através do repositório "scikit-feature" (LI et al., 2016). A ideia é verificar o comportamento da metodologia em conjunto de dados de diferentes domínios (ex.: Bioinformática e Processamento de Imagens), sendo três datasets de processamento de imagens e os outros cinco de bioinformática. A Tabela 2 mostra as características desses conjuntos de dados e também o tempo que foi preciso para a execução do MUI. Todos os conjuntos compartilham uma característica em comum, que é a de possuir o número de features (dimensões) muito maior do que o número de instâncias (objetos).

Tabela 2 – Conjuntos de Dados

Dataset	Domínio	Features	Instâncias	Classes	Tempo de Execução
ALLAML	Microarray	7129	72	2	9min
Carcinom	Microarray	9182	174	11	58min
Prostate-GE	Microarray	5966	102	2	20min
SMK-CAN	Microarray	19993	187	2	4h12min
TOX171	Microarray	5748	171	4	36min
pixraw10P	Imagem	10000	100	10	40min
warpAR10P	Imagem	2400	130	10	14min
warpPIE10P	Imagem	2420	210	10	20min

Para este projeto se utilizou um computador convencional com a seguinte configuração: processador Intel Core i7-3632QM (2.20GHz), 8GB de memória RAM, 500 GB de HD e o Ubuntu 16.04 LTS como sistema operacional. Observe na Tabela acima que (com exceção dos conjuntos SMK-CAN, Carcinom e pixraw10P) os tempos totais de execução obtidos não foram muito altos, isso quando analisamos a complexidade do trabalho em questão, porém em um trabalho futuro pode-se otimizar este tempo de execução, principalmente em conjuntos de dados onde a dimensionalidade esteja próximo ou superior a 10000 Features.

Os datasets de bioinformática são representados pelos microarrays, que são uma tecnologia que contém níveis de expressão de uma coleção de genes para uma dada amostra, também conhecido como chip de DNA ou microarranjos (HELLER, 2002). A tecnologia de microarrays permite a análise de centenas ou milhares de genes, em nosso exemplo, é possível observar pela Tabela 2 essa dimensionalidade, já os dados de imagem contém em seus objetos diferentes expressões faciais e como Features pixels em escala de cinza. Ambos os conjuntos de dados de microarray e imagens possuem variáveis contínuas.

Nas seções seguintes iremos detalhar os parâmetros utilizados por cada etapa do MUI para a execução do experimento.

4.1 NORMALIZAÇÃO

Para a execução do experimento, foi realizada a normalização dos dados, deixando-os em um mesmo intervalo, sendo os valores variando de 0 a 1. O método para esta tarefa foi o *straighforward normalization*, também utilizado no trabalho em (YAO et al., 2015). A Equação 4.1 detalha o processo de normalização. Onde cada Feature f e objetos i são representados em uma matriz de dados $n \times f$.

$$x_{if_1} = \frac{x_{if} - \min_{j=1}^{n} x_{jf}}{\max_{j=1}^{n} x_{jf} - \min_{j=1}^{n} x_{jf}} \tag{4.1}$$

Assim, a matriz de dados resultante deste passo servirá como entrada para o próximo passo, seleção prévia dos dados, onde as Features com baixa variância serão eliminados antes da execução da suíte de métodos de seleção de Features.

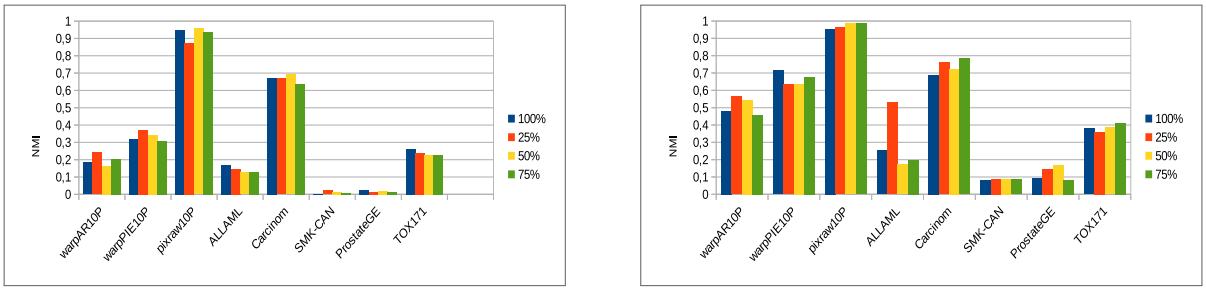
4.2 PRÉ-SELEÇÃO PELA VARIÂNCIA

Nesta seção realizamos uma análise exploratória para avaliar o interesse em fazer uma pré-seleção de features eliminando aquelas com baixa variância. O filtro de variância é uma forma simples e rápida de seleção de features, que poderia diminuir a complexidade dos datasets, contribuindo para uma maior rapidez na execução dos algoritmos de seleção de features na etapa posterior. Tal seleção tem o potencial de impactar negativamente, ou positivamente os resultados. Para verificar esses efeitos nos datasets explorados, realizamos experimentos com três limiares de variância: 25%, 50%, 75% e 100%. Esses limiares representam que percentagem do número de features deve ser mantido após o filtro da variância. Por exemplo, se utilizarmos o filtro da variância com o limiar de 75% temos o seguinte cenário: supondo que tenhamos um total inicial de 1000 features, calculamos a variância para cada uma dessas features (que já foram anteriormente normalizadas no mesmo intervalo), ordenamos pela variância de forma decrescente, e em seguida eliminamos as últimas 250 features da lista, resultando em um total final de 750 features, ou seja mantivermos 75% do total inicial de features. Note que 100% significa não eliminar nenhuma feature pelo filtro da variância.

Calculamos em seguida o NMI, Acurácia e Corrected Rand máximo obtido, considerando todas as configurações testadas, apenas utilizando o filtro de variância (chamado nos gráficos que seguem de "Apenas Filtro da Variância") e utilizando o filtro da variância como pré-seleção para os métodos de seleção de features mais sofisticados aqui considerados (chamado nos gráficos de "Com Seleção de Features").

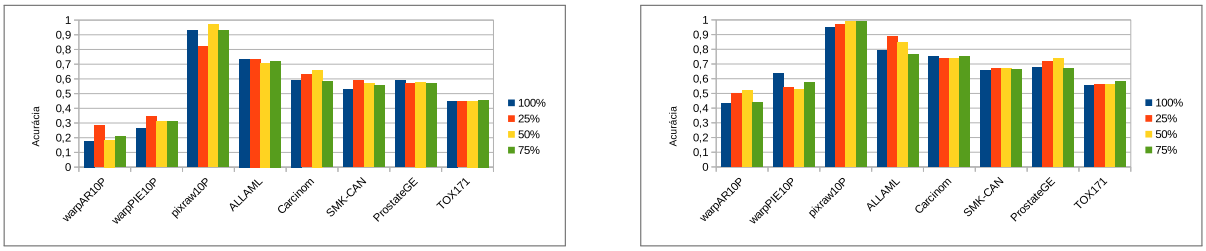
As Figuras 10, 11 e 12 mostram que, em geral, há de fato interesse em utilizar um método de seleção de features mais sofisticado do que apenas a seleção pela variância, o

que se reflete no fato de que os gráficos (b) nas três figuras (que usam métodos se seleção de features mais sofisticados) conseguem alcançar índices melhores do que os gráficos (a) para praticamente todos os conjuntos de dados (com exceção do conjunto de dados de imagem pixraw10P, que parece ser consideravelmente mais "fácil" do que os outros para a tarefa de agrupamento em relação a descobrir a partição previamente conhecida). No entanto, dentro do mesmo gráfico, notamos que há pouca mudança em relação ao uso dos diferentes níveis do filtro de variância como pré-filtragem (como ocorre em quase todos os conjuntos de dados). Podemos portanto inferir que as features eliminadas pela pré-filtragem tiveram pouco impacto (para melhor, ou pior) no resultado final. Dessa forma, em geral, não é de grande importância fazer uma pré-filtragem pelo filtro da variância quanto ao melhor resultado que pode ser obtido ao se aplicar posteriormente um método de seleção de features mais sofisticado. Por outro lado, eliminar um grande número de features como etapa inicial pode ser interessante para acelerar a execução dos métodos de seleção. Como todos os métodos utilizados nesse trabalho tiveram um tempo de execução pequeno, decidirmos não utilizar a pré-filtragem pela variância a fim de evitar ter mais um parâmetro para ser ajustado (que necessitaria na prática da utilização de um conjunto de validação). Assim a etapa de pré-seleção de features será desconsiderada nos nossos experimentos, ou seja, o conjunto de features utilizado para as próximas etapas será o conjunto com todas as features.



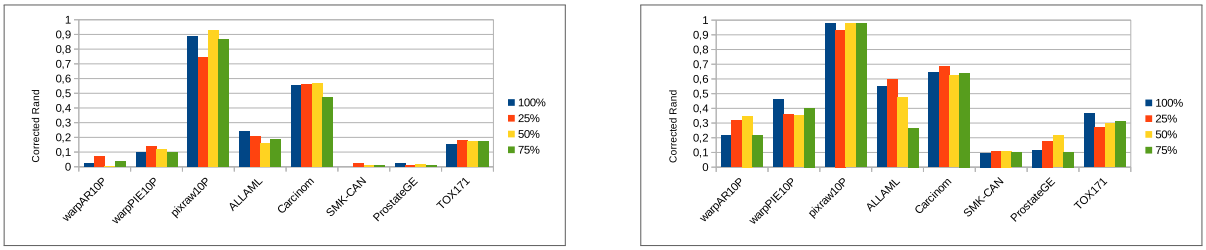
(a) Apenas Filtro da Variância (b) Com Seleção de Features

Figura 10 – NMI pelo Filtro da Variância



(a) Apenas Filtro da Variância (b) Com Seleção de Features

Figura 11 – Análise do Acurácia pelo Filtro da Variância



(a) Apenas Filtro da Variância

(b) Com Seleção de Features

Figura 12 – Análise do Corrected Rand pelo Filtro da Variância

4.3 GERAÇÃO DE MÚLTIPLOS RANKINGS

Como já mencionado na seção 3.3, iremos executar apenas os métodos e configurações listados na Tabela 1. As melhores configurações serão selecionadas a partir da média das silhuetas (maior valor) obtidas após a execução do k-means. Ao final teremos os 4 melhores resultados, um para cada método (LS, SPEC, GLSPFS e iDetect), e os resultados combinados entre os métodos através da contagem de Borda, a combinação será entre as melhores configurações obtidas por cada método, assim ao somar os resultados gerados a partir da execução isolada de cada método mais as junções pela contagem de Borda, teremos 15 possíveis resultados, sendo os quatro métodos isoladamente:

1. LS
2. SPEC
3. iDetect
4. GLSPFS, e mais 11 combinações de métodos, exaurindo todas as possibilidades de combinação entre eles,
5. LS e SPEC (LS-SPEC)
6. LS e iDetect (LS-iDetect)
7. LS e GLSPFS (LS-GLSPFS)
8. LS, SPEC e iDetect (LS-SPEC-iDetect)
9. LS, SPEC e GLSPFS (LS-SPEC-GLSPFS)
10. LS, iDetect e GLSPFS (LS-iDetect-GLSPFS)
11. SPEC e iDetect (SPEC-iDetect)
12. SPEC e GLSPFS (SPEC-GLSPFS)
13. SPEC, iDetect e GLSPFS (SPEC-iDetect-GLSPFS)

14. iDetect e GLSPFS (iDetect-GLSPFS)

15. LS, SPEC, iDetect e GLSPFS (LS-SPEC-iDetect-GLSPFS)

Esses resultados serão utilizados nas próximas etapas, onde serão gerados e avaliados subconjuntos de features para cada ranking.

4.4 GERAÇÃO DAS M-MELHORES FEATURES

Realizamos experimentos para analisar os resultados que potencialmente poderiam ser obtidos, para os conjuntos de dados considerados, ao se utilizar como método de definição do ponto de corte sobre o conjunto total de features o método do ponto de inflexão e o dos quartis, quando comparados com o que poderia ser obtido utilizando a estratégia de números mágicos. A ideia é que, se os métodos de seleção pelo ponto de inflexão e pelos quartis tiverem desempenho similar ou superior aos números mágicos, eles podem simplificar bastante a análise, uma vez que um grande número de pontos de corte considerados pelos números mágicos são reduzidos a apenas 4: ponto de maior inflexão e 1°, 2°, 3° quartis. Nesta análise não foi utilizada pré-filtragem pela variância. O índice considerado para medir o desempenho obtido foi o NMI. Os resultados apresentados consideram o valor máximo do NMI obtido para ponto de corte em questão considerando todas as possíveis configurações do método de seleção de features, por exemplo: considere que num determinado conjunto de dados tenhamos três rankings de features, um para cada uma das três configurações possíveis de um determinado método. Assim, iremos utilizar o valor máximo obtido para o NMI, obtido a partir do corte pela inflexão dos scores, entre os três rankings diferentes. Este valor máximo para o NMI obtido pela seleção do subconjunto de features através do ponto de inflexão, é então comparado com o máximo valor obtido para o NMI quando utilizamos a estratégia de ponto de corte por números mágicos. O mesmo processo é aplicado para o corte de features pelos quartis, considerando o primeiro, segundo e terceiro quartis. As Figuras 13 e 14 mostram os resultados obtidos.

Para a seleção do número de features pelos números mágicos, iremos iniciar com as 10 melhores features (conforme ordenação decrescente dos scores), em seguida faremos um incremento de mais 10 features até atingir um total de 200 features.

Observando os resultados exibidos nas figuras, vemos que para alguns conjuntos de dados as estratégias de ponto de corte pela inflexão e pelos quartis obtiveram resultados superiores aos números mágicos (ex.: uso dos quartis como ponto de corte no dataset Carcinom), porém em outros conjuntos o resultado foi inverso (ex.: uso do ponto de inflexão como ponto de corte no dataset warpAR10P). Dessa maneira, não podemos eliminar a estratégia de números mágicos para exploração de pontos de corte interessantes, entretanto as estratégias que propusemos, dos quartis e do ponto de inflexão, conseguem, por vezes, descobrir novos pontos de corte, que não foram explorados pela progressão aritmética dos números mágicos, que trazem resultados até melhores que todos aqueles

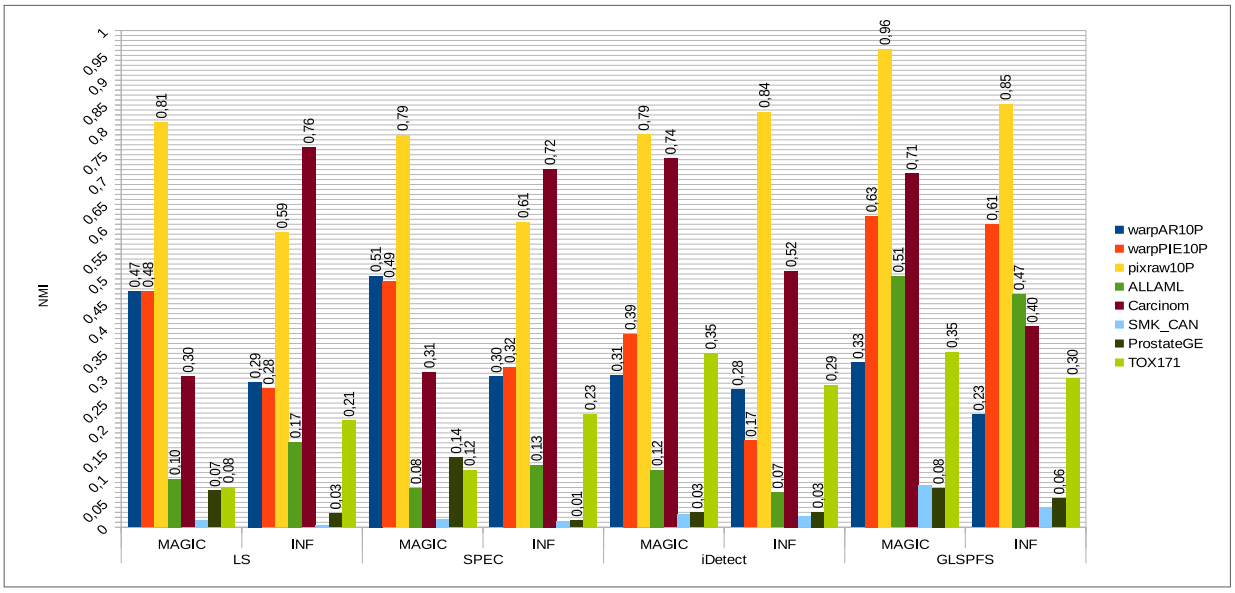


Figura 13 – Análise para a definição do ponto de corte no conjunto de features, comparando entre a estratégia de seleção a partir da inflexão dos scores (INF) e pelos números mágicos (MAGIC)

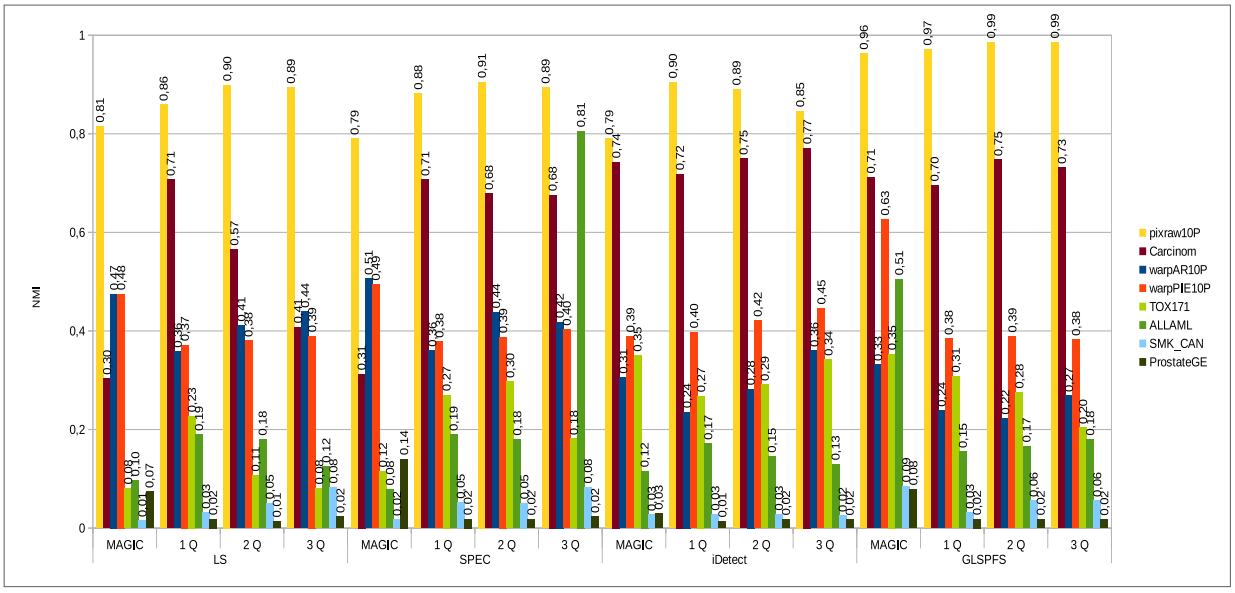


Figura 14 – Análise para a definição do ponto de corte no conjunto de features, comparando entre a estratégia de seleção a partir dos quartis (primeiro quartil - 1Q, segundo - 2Q e terceiro - 3Q) e os números mágicos (MAGIC)

explorados pelos números mágicos. Desta maneira iremos adotar essas estratégias de corte em adição às dos números mágicos. Em um trabalho futuro, seria interessante tratar o problema de reduzir a quantidade de pontos de corte a serem considerados.

4.5 AVALIAÇÃO DOS RESULTADOS

Iremos utilizar o critério não-supervisionado da média das silhuetas para selecionar os melhores subconjuntos de features resultantes das etapas anteriores. Serão listados os melhores resultados (maiores valores de silhueta) para cada método de seleção de features executado de forma isolada, bem como através da combinação entre os métodos pela contagem de Borda.

Nos conjuntos de dados aqui utilizados, há uma partição previamente conhecida dos objetos. Realizamos também uma comparação entre os clusters obtidos pelas configurações com melhor pontuação segundo nosso critério não-supervisionado e a partição previamente conhecida dos objetos. Serão utilizados os valores de acurácia (precisão), o NMI e o *Corrected Rand* (CR). Tais índices são amplamente utilizados na literatura para a comparação entre conjuntos. Note, entretanto, que algoritmos de cluster encontram "agrupamentos naturais" dos dados, que nem sempre correspondem às partições, que nesses datasets, são previamente conhecidas. Dessa maneira, essa forma de avaliar não julga necessariamente a "qualidade" dos clusters descobertos, apenas o quanto eles se assemelham a tais partições. Para uma avaliação real da qualidade dos clusters dentro da área do conhecimento (majoritariamente Biologia), seria necessária a consulta a especialistas, o que não foi possível no escopo desse trabalho. Para termos uma ideia do quanto os resultados gerados pela configuração escolhida com nosso critério não-supervisionado está próxima da configuração que geraria o resultado mais próximo da partição a priori, computamos os índices supervisionados para todas as configurações consideradas e registramos o máximo de cada um dos índices que foi obtido por alguma configuração, para comparação. Nas execuções do algoritmo de clustering, utilizamos o k-means clássico com o k igual ao número de classes previamente conhecidas e 50 como o total de inicializações aleatórias do k-means para a escolha dos centroides iniciais.

4.6 IMPLEMENTAÇÃO

O código de implementação de nossa metodologia foi escrito em Python (versão 3.5), para propósito de reprodução, o código completo está disponível em <https://github.com/marcosd3souza/FSMethodology>, onde foram utilizadas as implementações dos métodos de seleção de Features disponibilizadas publicamente na WEB. Para o iDetect foi utilizado o código disponibilizado em <https://www.acsu.buffalo.edu/~yijunsun/lab/iDetect.html>. Para o GLSPFS o código utilizado está disponível em <https://github.com/csliangdu/FSASL>. Os códigos de SPEC e Laplacian Score estão disponibilizados, como mencionado anteriormente, em "scikit-feature selection repository"(LI et al., 2016).

Apesar da implementação do MUI ter sido desenvolvida em Python, as implementações dos iDetect e GLSPFS estão disponibilizadas na plataforma MATLAB, assim foi utilizada

a biblioteca oct2py[1] para possibilitar a execução de código escrito em MATLAB a partir de uma seção de execução em Python. A portabilidade das implementações dos métodos de seleção de Features e do MUI para uma mesma linguagem ou plataforma poderia melhorar o tempo final de execução, além de remover a dependência com outras plataformas, isto pode ser um ponto de melhoria a ser desenvolvido em um trabalho futuro.

No apêndice deste trabalho também é possível obter um trecho, em Python, da metodologia MUI, a função *run_parallel_FS* é o método de entrada da metodologia, onde são feitas as iterações nos conjuntos de dados e nos métodos de seleção de Features, em seguida é feita a combinação de resultados utilizando a contagem de Borda.

[1] <https://pypi.org/project/oct2py/>

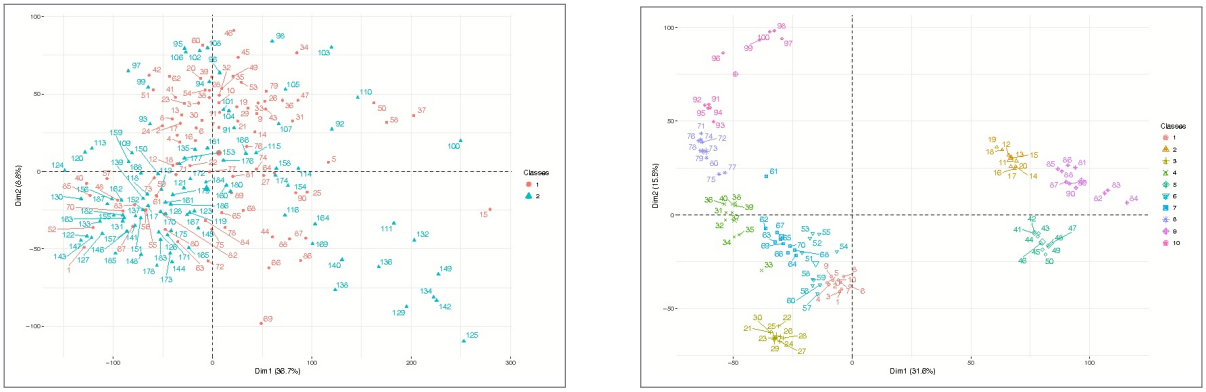
5 AVALIAÇÃO DOS RESULTADOS

Neste capítulo iremos analisar os resultados obtidos após a execução da metodologia proposta, porém antes é importante observar alguns cenários de referência. A Tabela 3 mostra o desempenho obtido em cada conjunto de dados ao se utilizar todas as features. Essa análise será nosso ponto de partida, sendo útil para compararmos os resultados obtidos ao se utilizar todas as features com aqueles utilizando o MUI. Nesta avaliação serão utilizados os seguintes indicadores: a média das silhuetas, o NMI, a acurácia e o Corrected Rand. Para a geração desses resultados, os dados utilizados foram normalizados de acordo com a etapa de normalização, proposta no Capítulo 4.

Tabela 3 – Resultados ao Utilizar Todas as Features

Dataset	**Silhueta**	**NMI**	**Acurácia**	**CR**
warpAR10P	0,19374	0,18329	0,17692	0,02626
warpPIE10P	0,20670	0,32143	0,26190	0,09848
pixraw10P	0,56971	0,94960	0,93000	0,88802
TOX171	0,10553	0,26377	0,45029	0,15370
ALLAML	0,20275	0,17135	0,73611	0,23795
Carcinom	0,17869	0,67127	0,59195	0,55523
SMK-CAN	0,41985	0,00175	0,52941	-0,00121
ProstateGE	0,64677	0,02550	0,58824	0,02276

Observe que os resultados obtidos pelos conjuntos de dados pixraw10P e Carcinom, ao se utilizar todas as features, já são bons, porém para os demais conjuntos os resultados obtidos são relativamente baixos. Os baixos valores obtidos para o NMI, acurácia e Corrected Rand significam que os clusters gerados tiveram baixa coincidência com as categorias previamente conhecidas dos objetos. Verificaremos se ao aplicar a metodologia MUI, proposta neste trabalho, obteremos resultados mais próximos às categorias previamente conhecidas em relação a quando se utiliza todas as Features. Também verificaremos se há um aumento da silhueta média, o que é um indicativo de melhores clusters segundo um critério não-supervisionado. Note que, se a performance obtida for próxima dos resultados ao se utilizarem todas as features, já podemos considerar um certo sucesso, pois possibilitou o mesmo desempenho com um conjunto de dimensionalidade menor em relação ao conjunto original de features.



(a) SMK-CAN (b) pixraw10P

Figura 15 – Análise de PCA Utilizando Todas as Features

Também foi realizada uma análise de PCA ao utilizar todas as features. A Figura 15 mostra[1], de forma visual, os objetos quando utilizamos os dois primeiros componentes. A análise de PCA nos mostra que o resultado obtido para o conjunto de dados SMK-CAN, assim como ocorre na maioria dos conjuntos de dados (exceto no pixraw10P), que não há uma separação clara entre os objetos de classes distintas quando usamos todas as features, isso fortalece a necessidade da utilização de métodos de seleção de features. Também é importante ressaltar que o MUI buscará gerar agrupamentos consistentes entre os objetos, através da silhueta, porém esses grupos podem não coincidir com as classes pré-conhecidas dos objetos, este fenômeno pode propor aos especialistas dos dados que classes distintas das previamente conhecidas podem existir.

Na Tabela 4 podemos observar os máximos valores obtidos para a acurácia, NMI e Corrected Rand nos conjuntos de dados considerando todas as configurações testadas. Estes resultados indicam os máximos valores que o MUI poderia alcançar, se conseguisse escolher, de forma não supervisionada, exatamente as melhores configurações segundo os critérios supervisionados. Por exemplo, no conjunto de dados warpAR10P o maior valor que poderia ser alcançado pela metodologia para o NMI é 0,48040, isso ocorreria se tivéssemos como utilizar o próprio NMI (maior valor) para escolher a melhor configuração, ao invés do nosso critério não supervisionado. Também repetimos este procedimento considerando a acurácia e o Corrected Rand como critério de escolha.

Como tratamos do caso não supervisionado, onde não dispomos da classe dos objetos, esses resultados servem apenas como referência para nossa metodologia, pois iremos utilizar a silhueta como critério de avaliação das features selecionadas. Em seguida iremos calcular o NMI, acurácia e Corrected Rand para essa configuração escolhida através da silhueta e comparar o desempenho obtido com os máximos valores listados na Tabela 4.

Observamos que realizando seleção de features é possível melhorar a qualidade dos

[1] Para fins de organização do documento, são exibidas apenas as análise visuais, neste capítulo, do pior desempenho (SMK-CAN) com o melhor desempenho (pixraw10P). No apêndice deste trabalho foram colocados os resultados para todos os conjuntos de dados.

Tabela 4 – Valores Máximos para NMI, Acurácia e Corrected Rand (CR).

Dataset	**NMI**	**Acurácia**	**CR**
warpAR10P	0,48040	0,49231	0,24917
warpPIE10P	0,68259	0,60476	0,45756
pixraw10P	0,98566	0,99000	0,97680
TOX171	0,37611	0,56140	0,25572
ALLAML	0,83644	0,97222	0,89002
Carcinom	0,78350	0,78736	0,69022
SMK-CAN	0,09836	0,67380	0,11620
ProstateGE	0,31896	0,79412	0,33996

clusters em todos os conjuntos de dados, mesmo naqueles onde já se obtém bons resultados sem qualquer seleção de features (ex.: pixraw10P). Isso pode ser verificado ao comparar os valores listados nas Tabelas 4 e 3.

A partir das Tabelas 3 e 4 podemos ter uma referência de performance para os oito conjuntos de dados. Os resultados mostram que existe alguma configuração, entre as consideradas, de algum método de seleção de features usado que consegue obter esse desempenho máximo. Usando a metodologia MUI, vamos listar a melhor configuração escolhida para cada método de seleção quando executado isoladamente (LS, SPEC, iDetect e GLSPFS) mais 11 possíveis combinações entre os métodos, ou seja uma lista com 15 opções de subconjuntos de features. De posse desta lista o especialista de dados poderá escolher qual subconjunto de features considera ser o mais interessante, ou então poderá ainda utilizar a "melhor escolha pelo MUI", correspondente à configuração, entre essas 15 opções, que tem o maior valor para o critério não supervisionado de escolha. Também podemos analisar quão boa seria a escolha realizada dessa forma.

A avaliação será separada por área de domínio, Bioinformática e Imagem. Esta divisão de análise se propõe a identificar possíveis comportamentos que possam surgir ao executar a metodologia em áreas de domínios diferentes.

5.1 RESULTADOS EM DADOS DE BIOINFORMÁTICA

Serão mostrados nesta seção os resultados obtidos nos conjuntos de dados de Bioinformática, onde serão descritas também as características de cada dataset. Esses conjuntos representam microarrays, onde os indivíduos (objetos) são descritos pelos genes (features). As features nesses conjuntos de dados, geralmente, são na ordem de milhares, enquanto que os objetos em dezenas ou centenas, assim nos deparamos com o clássico problema da maldição da dimensionalidade (DONOHO et al., 2000).

5.1.1 ALLAML

O ALLAML representa um microarray que foi gerado a partir de pacientes com câncer do tipo *acute myeloid leukemia* (AML) ou *acute lymphoblastic leukemia* (ALL), onde 47 desses estão classificados como ALL e 25 como AML. O conjunto de dados foi trazido no trabalho pioneiro de (GOLUB et al., 1999), que demonstra a capacidade de descobrir tipos derivados de câncer, bem como predizer os tipos já conhecidos, através do nível de expressões dos genes.

A fim de se obter uma visão sobre o desempenho da metodologia neste conjunto de dados, a Tabela 5 foi construída. Nesta Tabela pudemos observar todas as possibilidades de seleção de features disponibilizadas pelo MUI, os resultados são referentes a cada método executado de forma isolada e as combinações entre eles. Também é mostrado qual a abordagem para definir o número de features (quartis (Q1, Q2 e Q3), inflexão (INF) ou números mágicos (M)) e o número de features utilizadas (#fea), o valor obtido pela média das silhuetas, os resultados dos índices de NMI, acurácia e Corrected Rand obtidos pelo critério de avaliação do MUI (maior média das silhuetas) e qual valor máximo desses índices nos respectivos métodos. Essa comparação com os máximos valores dos índices são úteis para realizarmos uma breve verificação da diferença entre a escolha do MUI para o que seria considerado o melhor resultado de um método de seleção de features. Para uma fácil visualização desta análise iremos destacar os valores de interesse.

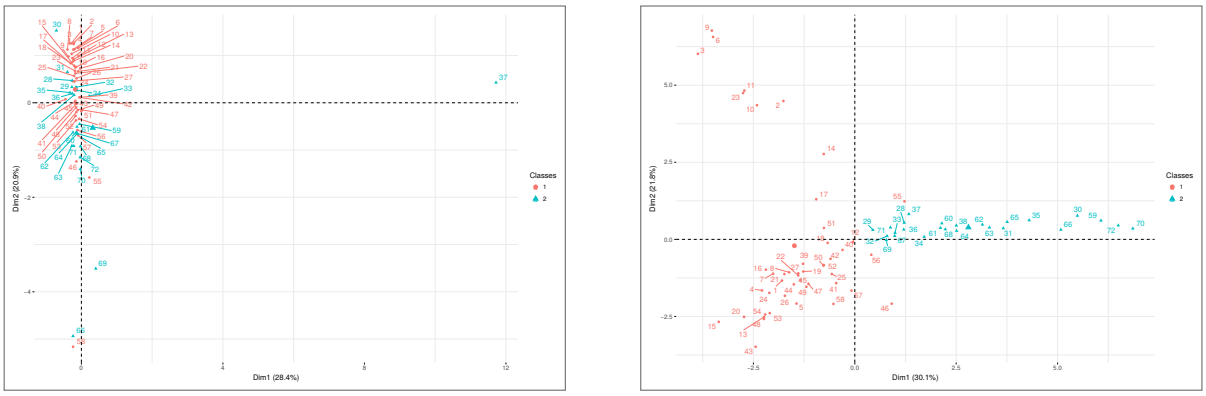
Observe na Tabela 5 que a melhor seleção de features, neste conjunto de dados, é a fornecida pela combinação SPEC-iDetect, onde se obteve os máximos valores para os índices. Porém a escolha pelo melhor índice segundo a metodologia (método SPEC), devido ao valor da silhueta ser superior, demonstra um desempenho para os índices supervisionados bastante inferior. Outras observações importantes, neste dataset, foi que a abordagem de escolha do número de features pelo ponto de inflexão proporcionou obter a maior silhueta e que as 10 features mais bem posicionadas no ranking de features, escolhidas pelo critério de números mágicos, aparecem como sugestão de ponto de corte (maior silhueta) na maioria dos casos. Outro resultado interessante é que a melhor configuração escolhida, de forma não-supervisionada pelo MUI, para o método SPEC-iDetect coincidiu com a melhor possível de ser obtida para esse método de seleção de features (note que os valores dos índices supervisionados e os valores max desses mesmos índices coincidem).

Tabela 5 – Resultados dos métodos utilizados para a seleção das features, do ponto de corte utilizado (cut), o número de features (#fea), silhueta, NMI, Corrected Rand (CR), Acurácia (Acc) e seus máximos valores no conjunto de dados ALLAML

Método	**cut**	**#fea**	**silhueta**	**nmi**	**max-nmi**	**acc**	**max-acc**	**CR**	**max CR**
LS	M	70	0,78967	0,04212	0,20942	0,62500	0,77778	-0,02421	0,29611
SPEC	INF	6	**0,88505**	0,06843	0,45114	0,66667	0,84722	0,02433	0,46463
iDetect	M	10	0,62157	0,07553	0,18951	0,68056	0,76389	0,11559	0,26626
GLSPFS	INF	5	0,50993	0,15807	0,46045	0,52778	0,87500	-0,04462	0,55278
LS-SPEC	M	10	0,71706	0,06598	0,14911	0,59722	0,63889	-0,04184	0,00939
LS-iDetect	M	20	0,32438	0,04853	0,62755	0,66667	0,93056	0,08993	0,73490
LS-GLSPFS	M	110	0,46324	0,04241	0,11155	0,68056	0,70833	0,07339	0,15246
LS-SPEC-iDetect	M	10	0,39711	0,08979	0,62755	0,70833	0,93056	0,11943	0,73651
LS-SPEC-GLSPFS	M	10	0,22353	0,00364	0,31532	0,54167	0,76389	-0,00615	0,26848
LS-iDetect-GLSPFS	M	10	0,29435	0,00033	0,45888	0,54167	0,81944	-0,01511	0,40037
SPEC-iDetect	M	20	**0,42274**	0,83644	0,83644	0,97222	0,97222	0,89002	0,89002
SPEC-GLSPFS	M	10	0,38464	0,30133	0,83644	0,69444	0,97222	0,13365	0,89002
SPEC-iDetect-GLSPFS	M	10	0,40872	0,33103	0,83644	0,72222	0,97222	0,18298	0,89002
iDetect-GLSPFS	M	10	0,34225	0,46815	0,61895	0,88889	0,90278	0,59552	0,64402
LS-SPEC-iDetect-GLSPFS	M	25	0,25807	0,43771	0,48132	0,80556	0,83333	0,36499	0,43722

De forma a termos uma ideia visual dos objetos descritos pelas features selecionadas, foi realizada uma análise de PCA sobre a escolha feita pelo MUI (maior silhueta) e o que pudemos considerar como o melhor resultado (segundo os índices supervisionados). Ou seja, os dados segundo as configurações em negrito da Tabela 5. Esta análise é útil para visualizarmos o comportamento dos objetos, considerando a utilização de apenas dois componentes principais (apenas duas dimensões) para cada uma das configurações.

Observe que no resultado de melhor silhueta, uma configuração do SPEC (cf. Figura 16a), os objetos se posicionaram mais próximos entre eles, e sem uma separação clara para os objetos que são de classes previamente conhecidas diferentes. Já os resultados da combinação entre os métodos SPEC e iDetect mostraram uma separação mais clara dos objetos entre suas categorias previamente conhecidas (Figura 16b). Também foi possível observar que o resultado obtido pelo SPEC-iDetect obteve um total de 51,9% (30,1 + 21,8) de variabilidade para os 2 primeiros componentes na execução do PCA, enquanto que o resultado utilizando apenas o SPEC sugerido pela metodologia obteve 49,3%.



(a) Seleção por SPEC

(b) Seleção por SPEC-iDetect

Figura 16 – Análise de PCA no Dataset ALLAML

5.1.2 CARCINOM

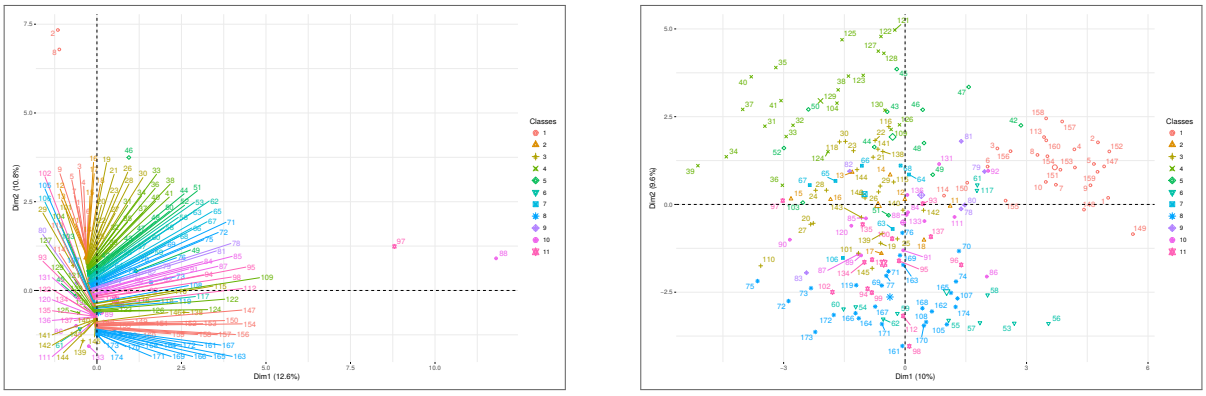
Este dataset contém 174 objetos que descrevem pacientes de 11 categorias, representando os seguintes tipos de câncer: próstata, uréter, mama, colorretal, gastroesofágico, rins, fígado, ovário, pâncreas, adenocarcinomas pulmonares, e carcinoma de células escamosas do pulmão, cada um deles possuindo, respectivamente, 26, 8, 26, 23, 12, 11, 7, 27, 6, 14, e 14 instâncias. Este conjunto de dados faz parte do trabalho realizado em (CAI et al., 2007), na qual propõe um método de seleção de genes mais relevantes. Os resultados da aplicação do MUI nesse conjunto de dados podem ser visualizados na Tabela 6.

Tabela 6 – Resultados dos métodos utilizados para a seleção das features, do ponto de corte utilizado (cut), o número de features (#fea), silhueta, NMI, Corrected Rand (CR), Acurácia (Acc) e seus máximos valores no conjunto de dados Carcinom

Método	**cut**	**#fea**	**silhueta**	**nmi**	**max-nmi**	**acc**	**max-acc**	**CR**	**max CR**
LS	M	10	0,45567	0,15940	0,66947	0,21839	0,67816	0,00917	0,55017
SPEC	M	10	**0,92499**	0,17570	0,75163	0,20690	0,74713	0,00900	0,67463
iDetect	INF	4	0,65143	0,42453	0,78350	0,39080	0,78736	0,20320	0,69022
GLSPFS	INF	4	0,53873	0,37373	0,76518	0,34483	0,78161	0,17610	0,67883
LS-SPEC	M	10	0,90275	0,19297	0,20522	0,20115	0,31034	0,01242	0,04598
LS-iDetect	M	10	0,18529	0,22777	0,45393	0,24713	0,54023	0,03909	0,30944
LS-GLSPFS	M	10	0,14973	0,16529	0,32786	0,22414	0,39655	0,01023	0,17518
LS-SPEC-iDetect	M	10	0,16694	0,22106	0,48904	0,28161	0,53448	0,05100	0,30879
LS-SPEC-GLSPFS	M	10	0,20342	0,16111	0,33469	0,20115	0,38506	0,00492	0,17071
LS-iDetect-GLSPFS	M	10	0,16928	0,24832	0,54290	0,23563	0,59770	0,05618	0,39872
SPEC-iDetect	M	45	**0,32837**	0,67499	0,73523	0,70115	0,75287	0,58572	0,66222
SPEC-GLSPFS	M	10	0,23836	0,18058	0,52534	0,25287	0,52299	0,02348	0,39174
SPEC-iDetect-GLSPFS	M	10	0,29483	0,40943	0,76446	0,41954	0,73563	0,21913	0,65444
iDetect-GLSPFS	M	15	0,35636	0,59854	0,77986	0,56322	0,74713	0,45793	0,66506
LS-SPEC-iDetect-GLSPFS	M	10	0,15375	0,25415	0,50128	0,28161	0,52299	0,08125	0,33649

Foi observado neste conjunto de dados que a seleção de features realizada pelo método escolhido através do MUI (SPEC), obteve um valor de silhueta bastante superior (0,92499), quando comparado com as demais opções também disponibilizadas. Porém a opção, entre as demais, que obteve o melhor desempenho para os índices supervisionados (NMI, acurácia e Corrected Rand) foi a junção entre os métodos SPEC e iDetect, como também ocorreu no conjunto de dados ALLAML. Os valores dos índices de NMI, acurácia e Corrected Rand para este conjunto de dados quando utilizamos todas as features foram: 0,67127, 0,59195 e 0,55523 respectivamente. Já ao utilizar as 45 features selecionadas pela junção entre SPEC e iDetect, o desempenho para o NMI (0,67499), a acurácia (0,70115) e Corrected Rand (0,58572) foi superior, ou seja, foi possível identificar uma configuração através da metodologia com melhores resultados, para os índices supervisionados, nessa atividade de agrupamento. Entretanto, como podemos observar nas colunas max-nmi, max-acc e max CR, existiam configurações com maiores valores para os índices supervisionados, mas que não foram selecionadas pelo método não-supervisionado.

Outra observação que se repetiu em ambos os conjuntos de dados (ALLAML e Carcinom), foi o desempenho dos critérios utilizados para a definição do número de features. Em ambos, o uso de números mágicos (com M=10) e o ponto de inflexão se repetiram como as melhores escolhas, ou seja, essas abordagens são responsáveis por obterem maiores valores de silhueta. Também vale destacar que o número de features selecionadas é consideravelmente menor do que o número original de features de ambos os conjuntos de dados (ALLAML e Carcinom), que possuíam, originalmente, 7129 e 9182 features, respectivamente, e após a execução da metodologia esse número foi reduzido para 20 e 45, respectivamente.



(a) Seleção por SPEC

(b) Seleção por SPEC-iDetect

Figura 17 – Análise de PCA no Dataset Carcinom

Através da análise PCA realizada entre as duas opções do MUI (SPEC e SPEC-iDetect), podemos verificar através da seleção de features pela opção SPEC-iDetect (Figura 17b), que apenas os objetos pertencentes a classe 1 são mais fáceis de serem agrupados, os demais objetos encontram-se muito próximos. Quando observamos o PCA executado

sobre as features selecionadas pelo método SPEC (Figura 17a), seleção cujo o resultado obteve maior silhueta, vemos que praticamente não há a separação entre os objetos. Vale salientar que em ambas análises de PCA, as variâncias representadas pelos dois primeiros componentes principais, quando somadas não totalizam 50% de variabilidade entre os objetos, o que sugere o uso de mais dimensões para melhor compreender o comportamento dos objetos. Porém esta análise será realizada utilizando apenas duas dimensões em todos os datasets, pois a utilizamos apenas para ter uma ideia visual em duas dimensões da projeção via PCA com os dois componentes principais, extraídos a partir das features selecionadas na configuração escolhida.

5.1.3 PROSTATE_GE

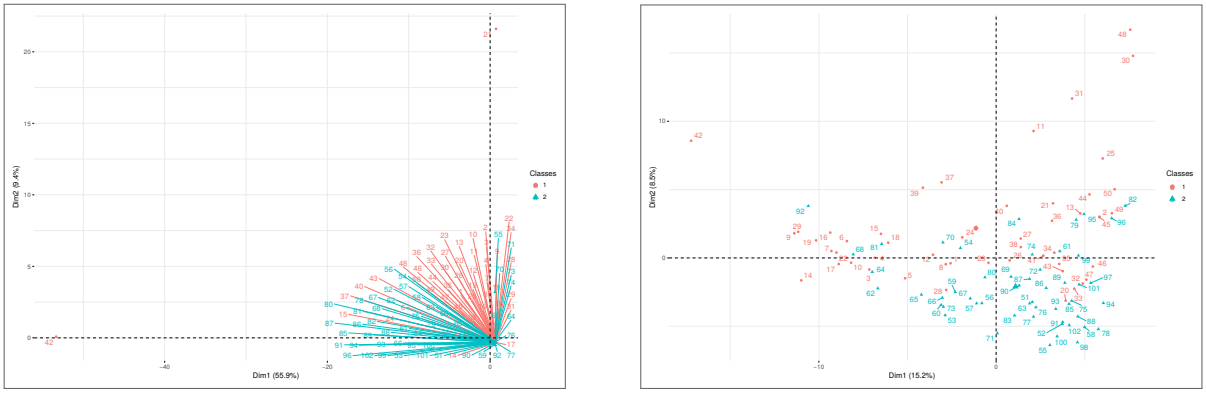
Este conjunto de dados é resultado do trabalho realizado em (SINGH et al., 2002), que busca classificar os objetos, representados por pacientes, que possuem determinado tumor e os que não possuem, categorizados como não-tumor. Os dados estão distribuídos da seguinte forma: 50 exemplos como não-tumor e 52 categorizados como pacientes com tumor. A Tabela 7 mostra os resultados obtidos após a execução do MUI.

O método escolhido pela metodologia, neste conjunto de dados, como também ocorre nos datasets anteriores, foi o SPEC, que obteve uma média de silhueta igual a 0,97148. A seleção das features realizada pelo SPEC foi responsáveis por obter o maior valor para o NMI, assim como também ocorre para as seleções realizadas pelos métodos LS e a combinação entre LS e SPEC. Também é importante ressaltar que a abordagem de seleção do número de features, neste conjunto de dados, foi através de números mágicos, abordagem essa que foi responsável por obter os maiores valores de silhueta.

Apesar da escolha feita pelo MUI apresentar o maior valor para o NMI entre as 15 configurações selecionadas pela metodologia, o desempenho máximo alcançável, em relação aos índices supervisionados, considerando todas as configurações é bem maior. Enquanto que as 50 features selecionadas pelo SPEC obteve um NMI de 0,03603, existe uma configuração que combina LS-SPEC-iDetect onde é possível alcançar um NMI igual a 0,31896. O que mostra que para esse conjunto de dados não houve uma boa coincidência entre o índice não-supervisionado e os índices supervisionados. A análise de PCA, na Figura 18, compara a configuração selecionada pelo MUI com aquela que alcançaria os maiores valores para os índices supervisionados.

Tabela 7 – Resultados dos métodos utilizados para a seleção das features, do ponto de corte utilizado (cut), o número de features (#fea), silhueta, NMI, Corrected Rand (CR), Acurácia (Acc) e seus máximos valores no conjunto de dados ProstateGE

Método	**cut**	**#fea**	**silhueta**	**nmi**	**max-nmi**	**acc**	**max-acc**	**CR**	**max CR**
LS	M	160	0,89078	0,03603	0,08503	0,51961	0,62745	0,00078	0,05855
SPEC	M	50	**0,97148**	0,03603	0,08503	0,51961	0,62745	0,00078	0,05855
iDetect	M	40	0,85350	0,01815	0,02550	0,57843	0,58824	0,01538	0,02276
GLSPFS	M	40	0,69248	0,01345	0,06358	0,56863	0,64706	0,00921	0,07738
LS-SPEC	M	105	0,94971	0,03603	0,03603	0,51961	0,51961	0,00078	0,00078
LS-iDetect	M	190	0,57382	0,01790	0,04118	0,57843	0,61765	0,01522	0,04627
LS-GLSPFS	M	10	0,19933	0,00004	0,07332	0,50000	0,65686	-0,00879	0,08963
LS-SPEC-iDetect	M	195	0,40314	0,01367	**0,31896**	0,56863	**0,79412**	0,00947	**0,33996**
LS-SPEC-GLSPFS	M	10	0,31096	0,00381	0,03123	0,51961	0,59804	-0,00464	0,03000
LS-iDetect-GLSPFS	M	185	0,54648	0,01773	0,02776	0,57843	0,59804	0,01509	0,02885
SPEC-iDetect	M	10	0,73058	0,02338	0,02338	0,58824	0,58824	0,02207	0,02207
SPEC-GLSPFS	M	10	0,35908	0,01793	0,05482	0,55882	0,61765	0,00646	0,04733
SPEC-iDetect-GLSPFS	M	195	0,73106	0,01345	0,03071	0,56863	0,59804	0,00921	0,02919
iDetect-GLSPFS	M	145	0,80777	0,01345	0,01345	0,56863	0,56863	0,00921	0,00921
LS-SPEC-iDetect-GLSPFS	M	120	0,42746	0,00982	0,13879	0,55882	0,66667	0,00422	0,10501



(a) Seleção por SPEC (b) Seleção por LS-SPEC-iDetect

Figura 18 – Análise de PCA no Dataset ProstateGE

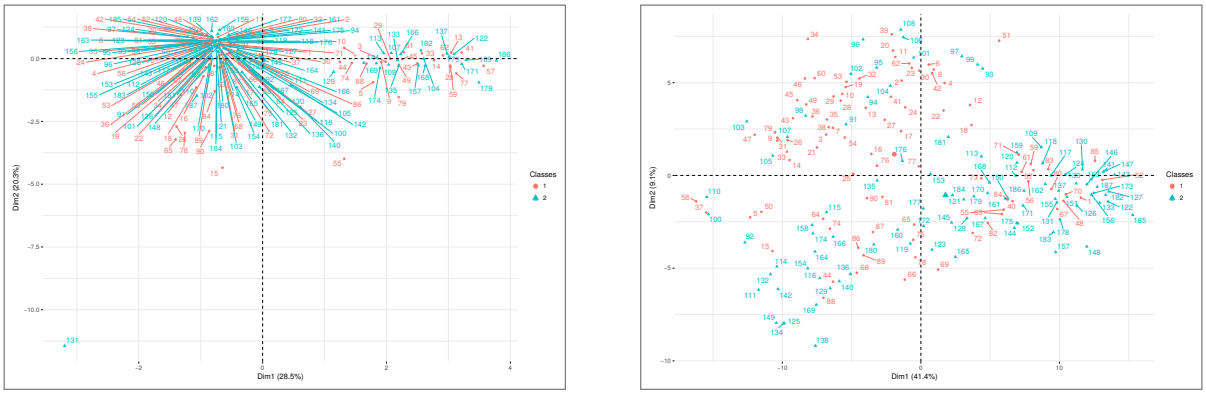
Na Figura 18a, ao utilizar as features selecionadas pelo método SPEC, observamos que não há uma separação clara entre os objetos de classes previamente conhecidas distintas, apesar das variâncias nos dois componentes principais, quando somadas, totalizarem 65,3%. Já a análise de PCA para a configuração que combina os métodos LS, SPEC e iDetect (Figura 18b) e possui os melhores índices supervisionados, mostra uma separação mais clara dos objetos para as duas classes. Porém em ambas as análises não há uma separação clara dos objetos, refletindo nos baixos valores obtidos pelos índices de NMI, acurácia e Corrected Rand neste conjunto de dados.

5.1.4 SMK-CAN

Este conjunto de dados trata-se do trabalho desenvolvido em (SPIRA et al., 2007) para a classificação de fumantes com ou sem câncer de pulmão. Os objetos estão distribuídos da seguinte forma: 92 pacientes (52%) não diagnosticados com o câncer, e 95 pacientes (48%) diagnosticados com câncer. Note que a expectativa é a geração de dois agrupamentos com quantidade de objetos bem equilibrada. Este dataset, entretanto, possui a maior dimensionalidade (19993 Features) entre os testados, aumentando nesse sentido a dificuldade para a seleção de features e geração de agrupamentos relevantes.

Na Tabela 8 é possível observar, de forma geral, que todas as opções de seleção de features disponibilizadas pelo MUI obtiveram baixos desempenhos nos índices supervisionados NMI e Corrected Rand. Além disso, os valores máximos alcançáveis para os índices supervisionados foram baixos, mostrando que é muito difícil encontrar as classes previamente conhecidas como agrupamentos naturais nesse conjunto de dados. A melhor configuração apresentada pela metodologia para o NMI tratou-se da combinação iDetect-GLSPFS. Desta forma iremos comparar os resultados obtidos pela escolha do MUI (SPEC) com o iDetect-GLSPFS, na Figura 19. Podemos observar também que na execução da metodologia para os métodos isoladamente, o ponto de inflexão como estratégia para escolher o ponto de corte do número de features teve melhor desempenho não supervisionado, ou seja, esta

abordagem foi responsável por obter os maiores valores para a média das silhuetas entre os objetos.



(a) Seleção por SPEC

(b) Seleção por iDetect-GLSPFS

Figura 19 – Análise de PCA no Dataset SMK-CAN

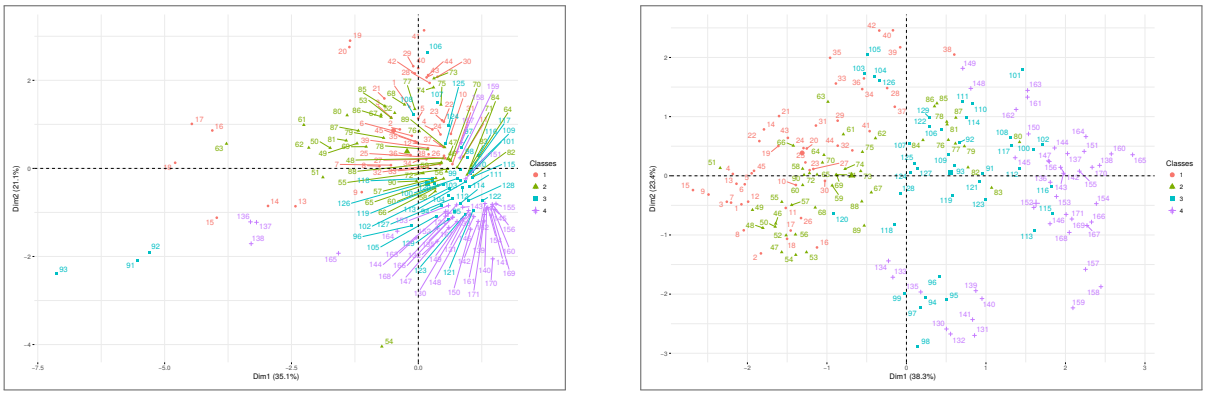
Através da análise de PCA, foi possível observar que a seleção de features realizada pela opção iDetect-GLSPFS proporcionou, através de dois componentes principais, uma variabilidade total de 96,1% para os objetos. Podemos considerar desta forma que esta visualização mostra de forma bastante suficiente, o posicionamento dos objetos em um espaço bidimensional. Porém podemos notar que para ambos os casos comparados na Figura 19, é difícil separar os objetos em relação às classes previamente conhecidas, pois em ambas análises não é possível visualizar, de forma clara, a formação de dois agrupamentos referentes a essas classes.

Tabela 8 – Resultados dos métodos utilizados para a seleção das features, do ponto de corte utilizado (cut), o número de features (#fea), silhueta, NMI, Corrected Rand (CR), Acurácia (Acc) e seus máximos valores no conjunto de dados SMK-CAN

Método	**cut**	**#fea**	**silhueta**	**nmi**	**max-nmi**	**acc**	**max-acc**	**CR**	**max CR**
LS	INF	4	0,51078	0,00778	0,01438	0,52941	0,57219	-0,00066	0,01577
SPEC	INF	6	**0,70266**	0,00091	0,03828	0,52406	0,60963	-0,00199	0,04351
iDetect	INF	4	0,59708	0,02197	0,02482	0,58824	0,59358	0,02610	0,02999
GLSPFS	INF	4	0,56439	0,03093	0,09836	0,60428	0,67380	0,03844	0,11620
LS-SPEC	M	10	0,51722	0,00091	0,00954	0,52406	0,55615	-0,00199	0,00760
LS-iDetect	M	10	0,29032	0,01766	0,03836	0,57754	0,61497	0,01931	0,04776
LS-GLSPFS	M	15	0,30711	0,00000	0,04957	0,50802	0,63102	-0,00427	0,06372
LS-SPEC-iDetect	M	10	0,29748	0,02346	0,02921	0,58824	0,59893	0,02644	0,03437
LS-SPEC-GLSPFS	M	35	0,31309	0,00037	0,03843	0,51872	0,61497	-0,00296	0,04796
LS-iDetect-GLSPFS	M	20	0,30869	0,00160	0,09017	0,51337	0,67380	-0,00403	0,11608
SPEC-iDetect	M	180	0,33155	0,03932	0,05404	0,61497	0,63102	0,04777	0,06371
SPEC-GLSPFS	M	10	0,23189	0,03910	0,04244	0,61497	0,62032	0,04805	0,05304
SPEC-iDetect-GLSPFS	M	10	0,33890	0,02061	0,04867	0,58289	0,62567	0,02279	0,05816
iDetect-GLSPFS	M	150	**0,40356**	0,05015	0,05492	0,62567	0,63102	0,05820	0,06374
LS-SPEC-iDetect-GLSPFS	M	15	0,36575	0,00092	0,07264	0,50802	0,65775	-0,00446	0,09469

5.1.5 TOX171

Este conjunto de dados trata-se de um experimento realizado em ratos, onde se tenta classificar os indivíduos, após a ingestão de algumas substâncias químicas, em quatro classes (PILOTO; SCHILLING, 2010). A Tabela 9 mostra os resultados obtidos após a execução do MUI. O desempenho neste dataset, ao se utilizar todas as features, para os índices supervisionados foi superior quando comparamos com as opções disponibilizadas através da execução do MUI, o que indica que não houve uma boa coincidência entre o índice não supervisionado e os supervisionados. Porém ao observarmos os máximos valores que poderiam ser alcançados pelos métodos iDetect e GLSPFS, é possível notar que há configurações para os métodos de seleção que conseguiriam obter um desempenho superior em relação aos índices supervisionados, quando comparada com os resultados obtidos ao se utilizar todas as features.



(a) Seleção por SPEC

(b) Seleção por iDetect

Figura 20 – Análise de PCA no Dataset TOX171

Através da análise de PCA sobre as features selecionadas pelo método iDetect, podemos observar, através da Figura 20b, que não há uma formação clara de agrupamento para os objetos da classe 3, é possível sugerir que os objetos das classes 1 e 2 possam formar um único cluster e que os objetos da classe 4 formam um cluster mais claro. Já referente à análise de PCA a partir das features selecionadas pelo método SPEC (sugerido pelo MUI) é possível notar uma separação mais clara entre os objetos das classes 1, 3 e 4.

Tabela 9 – Resultados dos métodos utilizados para a seleção das features, do ponto de corte utilizado (cut), o número de features (#fea), silhueta, NMI, Corrected Rand (CR), Acurácia (Acc) e seus máximos valores no conjunto de dados TOX171

Método	**cut**	**#fea**	**silhueta**	**nmi**	**max-nmi**	**acc**	**max-acc**	**CR**	**max CR**
LS	INF	4	0,20269	0,02531	0,23605	0,33333	0,49708	0,00675	0,18123
SPEC	INF	5	**0,54547**	0,05379	0,27308	0,36842	0,46199	0,02666	0,19509
iDetect	INF	4	**0,31608**	0,15558	0,37611	0,42105	0,50292	0,10472	0,25517
GLSPFS	INF	4	0,34807	0,04892	0,35222	0,33918	0,52047	0,03049	0,21689
LS-SPEC	10	10	0,16377	0,01383	0,06986	0,29825	0,37427	-0,00528	0,03900
LS-iDetect	10	10	0,15860	0,03758	0,15609	0,32164	0,44444	0,01722	0,10520
LS-GLSPFS	10	10	0,08816	0,01938	0,07705	0,32164	0,36257	0,00009	0,04636
LS-SPEC-iDetect	10	10	0,15222	0,04040	0,12174	0,32749	0,43275	0,01732	0,08804
LS-SPEC-GLSPFS	10	10	0,09607	0,01854	0,07080	0,29825	0,38012	-0,00146	0,04002
LS-iDetect-GLSPFS	10	10	0,14473	0,04027	0,18370	0,32749	0,42105	0,01805	0,10294
SPEC-iDetect	10	10	0,17102	0,08167	0,17117	0,36257	0,40936	0,04305	0,12129
SPEC-GLSPFS	10	10	0,12546	0,03549	0,09538	0,33918	0,39181	0,01145	0,05333
SPEC-iDetect-GLSPFS	10	10	0,15534	0,04208	0,20786	0,33333	0,47953	0,01479	0,15682
iDetect-GLSPFS	10	10	0,13846	0,09928	0,33067	0,43275	0,56140	0,07650	0,25572
LS-SPEC-iDetect-GLSPFS	10	10	0,15002	0,03201	0,14400	0,31579	0,43275	0,01084	0,09540

5.1.6 RESUMO DOS RESULTADOS NOS DATASETS DE BIOINFORMÁTICA

Pelos resultados obtidos nos conjunto de dados de Bioinformática, pudemos observar que a seleção de features a partir do método SPEC foi responsável por obter os maiores valores de silhueta. Também pudemos notar que os números mágicos, e a abordagem pelo ponto de inflexão foram responsáveis, através do critério de maior silhueta, pelas melhores configurações em todos os casos. Também vale destacar que a combinação entre os resultados dos métodos de seleção de features, através da contagem de Borda, possibilitou a obtenção dos melhores desempenhos para os índices supervisionados (NMI, acurácia e Corrected Rand), em todos os datasets, exceto para o dataset TOX171 onde a melhor opção foi a partir da seleção realizada pelo método iDetect, sinalizando um real interesse em combinar os resultados dos métodos de seleção.

Através das análises de PCA realizadas no conjuntos de dados de Bioinformática, pudemos observar que na maioria dos casos não se observou uma formação clara de clusters coincidentes com as categorias previamente conhecidas dos objetos.

5.2 RESULTADOS EM DADOS DE IMAGEM

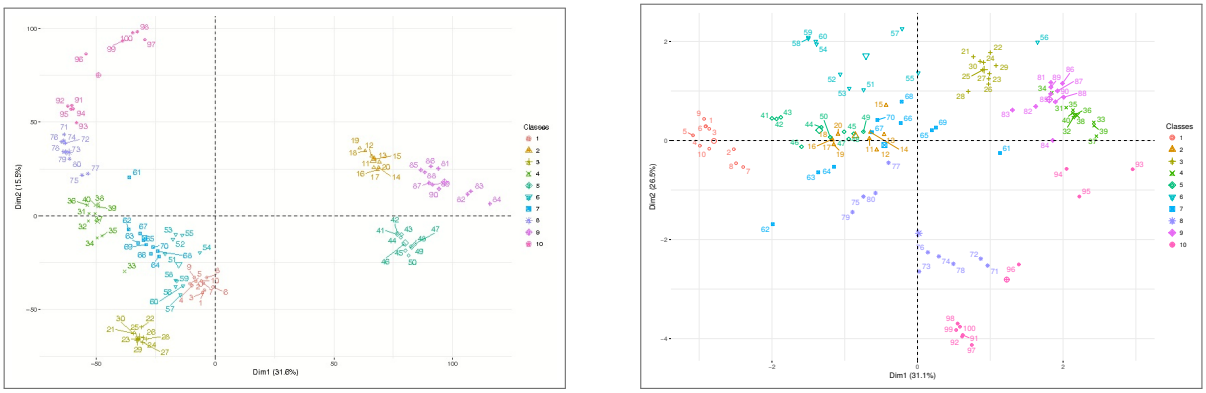
A análise dos resultados no conjuntos de dados de imagem se refere à atividade de reconhecimento de faces, onde os objetos representam uma determinada expressão facial e as features os pixels para cada imagem. Através dos estudos nestes conjuntos de dados a ideia se resume em reconhecer as faces em determinadas expressões ou poses a partir dos pixels da imagem.

5.2.1 PIXRAW10P

Este conjunto de dados contém inicialmente 10000 pixels para cada imagem, ao total são 100 imagens que representam 10 categorias de faces, que podem ser mudanças de expressões e poses. Este dataset pode ser obtido a partir do trabalho realizado em (HOND; SPACEK, 1997). Neste trabalho os pixels serão as features, cada imagem será uma instância (objeto) e as 10 categorias das imagens serão as classes dos objetos. A Tabela 10 mostra os resultados obtidos após a execução do MUI neste conjunto de dados.

Observe que a escolha sugerida pelo MUI, a partir da silhueta, conseguiu alcançar o melhor resultado, entre as 14 outras opções apresentadas. Outro ponto importante a se destacar foi que a abordagem pelo ponto de inflexão, selecionando as 7 melhores features, foi responsável por essa seleção. Porém havia uma outra configuração do GLSPFS que teria alcançado um NMI igual a 0,98566. Note que, como foi visto anteriormente, o uso de todas as features neste conjunto de dados já alcançava um excelente resultado em relação aos índices supervisionados, com 0,94960 para o NMI, 0,93000 para a acurácia e 0,88802 para o Corrected Rand. Desta maneira a configuração escolhida pelo MUI teve um desempenho inferior para os índices supervisionados em relação a configuração com todas

as features. Entretanto, podemos considerar que os índices alcançados pela configuração selecionada (0,807304 para NMI, 0,85000 para acurácia e 0,74709 para o CR) ainda são bons, e teve-se uma grande compressão dos dados, passando de 10000 para apenas 7 features. Na Figura 21 comparamos a projeção com os dois componentes principais da análise de PCA da configuração do GLSPFS selecionada pelo MUI e aquela com todas as features.



(a) Utilizando todas as features

(b) Seleção por GLSPFS

Figura 21 – Análise de PCA no Dataset pixraw10P

Na figura é possível observar que ao utilizarmos todas as features há uma separação mais clara dos objetos, considerando as classes previamente conhecidas. Porém ao somarmos as variâncias dos dois componentes principais, temos um total de 47,1%. Ao realizarmos a análise de PCA para as features selecionadas pelo método GLSPFS, sugerido pelo MUI, obtemos na soma das variâncias dos dois componentes um total de 96%. Vale salientar que em ambos os casos, os objetos de uma mesma classe se encontram próximos, o que se reflete nos bons índices alcançados. Podemos considerar com isso que este dataset apresenta menor complexidade para a tarefa de agrupamento do que os outros estudados.

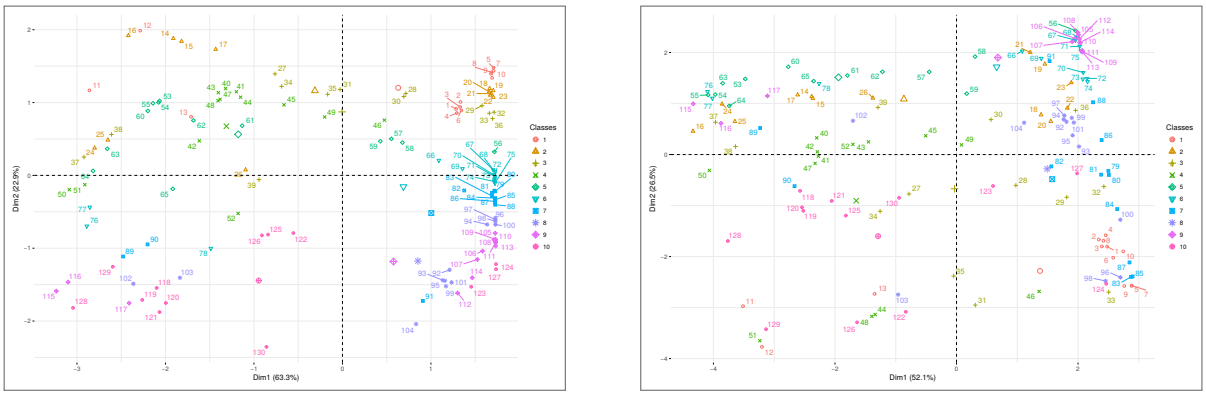
Tabela 10 – Resultados dos métodos utilizados para a seleção das features, do ponto de corte utilizado (cut), o número de features (#fea), silhueta, NMI, Corrected Rand (CR), Acurácia (Acc) e seus máximos valores no conjunto de dados pixraw10P

Método	**cut**	**#fea**	**silhueta**	**nmi**	**max-nmi**	**acc**	**max-acc**	**CR**	**max CR**
LS	M	90	0,61435	0,66367	0,93970	0,45000	0,84000	0,34650	0,83643
SPEC	M	90	0,61435	0,66367	0,93970	0,45000	0,84000	0,34650	0,83643
iDetect	M	120	0,62768	0,77487	0,94291	0,66000	0,95000	0,52604	0,89351
GLSPFS	**INF**	7	**0,70702**	0,87304	0,98566	0,85000	0,99000	0,74709	0,97680
LS-SPEC	M	90	0,61435	0,66367	0,66910	0,45000	0,56000	0,34650	0,39339
LS-iDetect	M	15	0,58073	0,63112	0,75687	0,48000	0,72000	0,29577	0,54647
LS-GLSPFS	M	10	0,61194	0,67710	0,84300	0,51000	0,78000	0,31031	0,67511
LS-SPEC-iDetect	M	10	0,54442	0,64293	0,73806	0,55000	0,68000	0,37850	0,48900
LS-SPEC-GLSPFS	M	10	0,46510	0,64860	0,82608	0,56000	0,76000	0,35603	0,65601
LS-iDetect-GLSPFS	M	10	0,54584	0,70780	0,87828	0,60000	0,78000	0,47147	0,70754
SPEC-iDetect	M	15	0,58073	0,63112	0,75687	0,48000	0,72000	0,29577	0,54647
SPEC-GLSPFS	M	10	0,61194	0,67710	0,84300	0,51000	0,78000	0,31031	0,67511
SPEC-iDetect-GLSPFS	M	10	0,54584	0,70780	0,87828	0,60000	0,78000	0,47147	0,70754
iDetect-GLSPFS	M	10	0,67713	0,79340	0,96302	0,70000	0,97000	0,55261	0,93311
LS-SPEC-iDetect-GLSPFS	M	10	0,54073	0,73365	0,78634	0,60000	0,72000	0,50084	0,59771

5.2.2 WARPAR10P

Este conjunto de dados contém imagens com oclusões de ambiente, mudanças de iluminação e expressões faciais. O total de pixels por imagem neste conjunto é 2400. As classes estão dispostas em 10 categorias e os objetos são representados pelo total de 130 imagens. A Tabela 11 mostra os resultados obtidos após a execução do MUI para este conjunto de dados.

Observe que sem a utilização de qualquer método de seleção de features, para este dataset, é obtido o seguinte desempenho: 0,18329 para o NMI, 0,17692 para a acurácia e 0,02626 para o Corrected Rand. Quando utilizamos a seleção de features através do método iDetect (sugerida pelo MUI), é possível observar que o desempenho obtido é superior quando comparamos ao resultado na utilização de todas as features. O desempenho pela seleção através do método iDetect foi: 0,32465 para o NMI, 0,31538 para a acurácia e 0,08163 para o Corrected Rand. Porém através da seleção pela combinação entre os métodos SPEC e iDetect é possível obter um resultado ainda melhor. O resultado obtido pelo método SPEC, ao ser combinado com o resultado pelo método iDetect, contribui para o que seria considerado como o melhor resultado entre as opções disponibilizadas pelo MUI.



(a) Seleção por iDetect

(b) Seleção por SPEC-iDetect

Figura 22 – Análise de PCA no Dataset warpAR10P

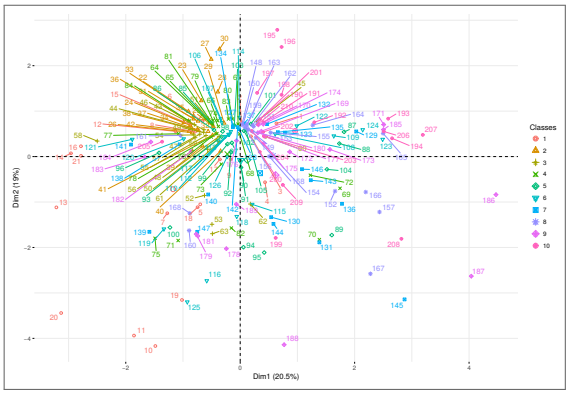
Através da análise de PCA, é possível observar através das features selecionadas pelo método iDetect, que é possível identificar agrupamentos entre objetos de classes distintas, pois é notável perceber a aproximação dos objetos nesses clusters. Este fato pode sugerir a especialistas de dados o surgimento de uma nova categoria, ou a possível alteração da classe dos objetos. Também vale ressaltar que as features selecionadas pelo método iDetect possibilitou uma aproximação maior entre os objetos quando comparamos pela análise de PCA realizada sobre as features selecionadas pela combinação entre os métodos SPEC e iDetect.

Tabela 11 – Resultados dos métodos utilizados para a seleção das features, do ponto de corte utilizado (cut), o número de features (#fea), silhueta, NMI, Corrected Rand (CR), Acurácia (Acc) e seus máximos valores no conjunto de dados warpAR10P

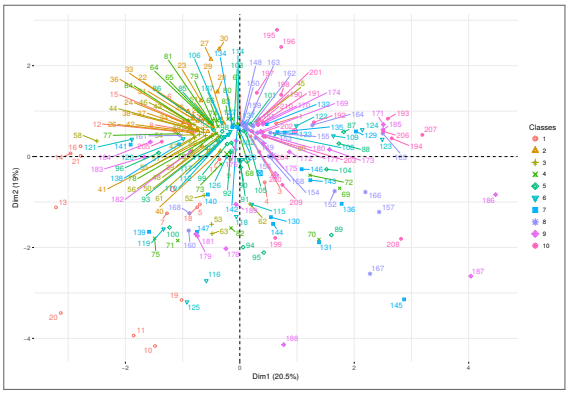
Método	**cut**	**#fea**	**silhueta**	**nmi**	**max-nmi**	**acc**	**max-acc**	**CR**	**max CR**
LS	INF	4	0,35220	0,27753	0,42943	0,29231	0,42308	0,05494	0,20342
SPEC	M	10	0,48261	0,36147	0,46799	0,32308	0,44615	0,07061	0,21210
iDetect	INF	4	**0,62347**	0,32465	0,33353	0,31538	0,36154	0,08163	0,11916
GLSPFS	INF	4	0,53159	0,22735	0,28182	0,26154	0,32308	0,03797	0,08516
LS-SPEC	M	10	0,47022	0,35517	0,48040	0,27692	0,46923	0,08173	0,24917
LS-iDetect	M	10	0,39839	0,39664	0,44266	0,43846	0,43846	0,17818	0,20459
LS-GLSPFS	M	10	0,20853	0,25204	0,36389	0,30769	0,36154	0,05634	0,12483
LS-SPEC-iDetect	M	10	0,37405	0,29344	0,45176	0,31538	0,49231	0,07793	0,23531
LS-SPEC-GLSPFS	M	10	0,23206	0,33320	0,40716	0,33846	0,36923	0,09535	0,13509
LS-iDetect-GLSPFS	M	10	0,37061	0,35512	0,43296	0,36154	0,44615	0,11441	0,16885
SPEC-iDetect	M	15	**0,34298**	0,43040	0,45375	0,43077	0,44615	0,21135	0,22513
SPEC-GLSPFS	M	10	0,34175	0,34322	0,37569	0,34615	0,38462	0,08382	0,13017
SPEC-iDetect-GLSPFS	M	10	0,36625	0,37561	0,37561	0,36154	0,36154	0,12423	0,12423
iDetect-GLSPFS	M	10	0,45911	0,24489	0,30538	0,30000	0,35385	0,05323	0,09536
LS-SPEC-iDetect-GLSPFS	M	10	0,37120	0,33139	0,38926	0,33846	0,41538	0,10028	0,16302

5.2.3 WARPPIE10P

Este dataset contém significantes variações de iluminação na captura de algumas imagens, onde cada imagem possui um um total de 2420 pixels. A Tabela 12 mostra os resultados obtidos pelo MUI para este conjunto de dados. Observe que as opções que obtiveram os maiores valores para a silhueta, usando os métodos LS e SPEC, tiveram exatamente o mesmo desempenho, além de que o número de features escolhido para ambos os métodos também foi igual, 6 features, selecionadas pelo ponto de inflexão. Desta forma iremos realizar a comparação entre as opções: LS, SPEC e iDetect (a opção que obteve maior valor para o NMI).



(a) Seleção por LS



(b) Seleção por SPEC

(c) Seleção por iDetect

Figura 23 – Análise de PCA no Dataset warpPIE10P

Através da análise de PCA para as três opções (LS, SPEC e iDetect), pudemos observar que as features utilizadas pelos métodos LS e SPEC são as mesmas, pois apresentam um mesmo resultado a partir dos dois componentes principais. Em relação a seleção de features a partir do método iDetect, podemos concluir que apesar de ter apresentado o melhor NMI (0,42390) e as variâncias dos dois componentes principais quando somadas totalizem 83,6%, não é possível visualizar de forma clara a separação dos objetos em suas respectivas classes, ou a possível formação de clusters para objetos de uma mesma classe, com exceção para os objetos pertencentes às classes 2, 4 e 10.

Tabela 12 – Resultados dos métodos utilizados para a seleção das features, do ponto de corte utilizado (cut), o número de features (#fea), silhueta, NMI, Corrected Rand (CR), Acurácia (Acc) e seus máximos valores no conjunto de dados warpPIE10P

Método	**cut**	**#fea**	**silhueta**	**nmi**	**max-nmi**	**acc**	**max-acc**	**CR**	**max CR**
LS	INF	6	**0,62572**	0,15623	0,68259	0,20952	0,60476	0,01034	0,45756
SPEC	INF	6	**0,62572**	0,15623	0,68259	0,20952	0,60476	0,01034	0,45756
iDetect	M	140	**0,37137**	**0,42390**	0,44567	0,42857	0,44286	0,14778	0,17056
GLSPFS	INF	4	0,33718	0,26469	0,42389	0,29048	0,41429	0,07355	0,16087
LS-SPEC	M	10	0,41122	0,21376	0,54832	0,24286	0,52857	0,03354	0,30545
LS-iDetect	M	10	0,24303	0,29586	0,38538	0,29524	0,35238	0,09547	0,13853
LS-GLSPFS	M	10	0,28460	0,19789	0,53168	0,24762	0,52857	0,04718	0,29623
LS-SPEC-iDetect	M	10	0,26813	0,31780	0,37764	0,31905	0,39524	0,10983	0,14133
LS-SPEC-GLSPFS	M	10	0,31658	0,23416	0,56965	0,25714	0,57619	0,06517	0,28654
LS-iDetect-GLSPFS	M	15	0,29147	0,23748	0,38102	0,23810	0,34286	0,04447	0,11000
SPEC-iDetect	M	10	0,24303	0,29586	0,38538	0,29524	0,35238	0,09547	0,13853
SPEC-GLSPFS	M	10	0,28460	0,19789	0,53168	0,24762	0,52857	0,04718	0,29623
SPEC-iDetect-GLSPFS	M	15	0,29147	0,23748	0,38102	0,23810	0,34286	0,04447	0,11000
iDetect-GLSPFS	M	85	0,32750	0,37079	0,38730	0,34762	0,35238	0,10930	0,15013
LS-SPEC-iDetect-GLSPFS	M	190	0,22937	0,30966	0,38126	0,35714	0,36667	0,07772	0,13124

Nos conjuntos de dados de imagem observou-se que através da análise de PCA, utilizando os dois componentes principais, foram reveladas estruturas mais claras de agrupamento quando comparamos com datasets de bioinformática, possibilitando que os resultados apresentados sejam, de modo geral, melhores. Nota-se que a complexidade dos datasets de bioinformática mostrou-se superior aos datasets de imagem. Outro fator importante foi que o método iDetect apresentou um bom desempenho para este tipo de domínio, isso quando a seleção de features foi realizada a partir da execução isolada do método ou a partir da combinação com o método SPEC. Também vale ressaltar que a abordagem do ponto de inflexão obteve os maiores valores de silhueta, juntamente com os números mágicos.

5.3 RESUMO DOS RESULTADOS

Através da Tabela 13 podemos comparar o desempenho de nossa metodologia. Iniciaremos com o conjunto de dados ALLAML, onde sem a utilização de qualquer método de seleção de features, os valores calculados para o NMI, acurácia e Corrected Rand foram $0,17135$, $0,73611$ e $0,23795$, respectivamente, conforme foi listado anteriormente na Tabela 3. Após realizarmos a seleção de features, através da configuração utilizada na fusão entre os rankings dos métodos SPEC e iDetect pudemos obter os valores máximos para o NMI, acurácia e Corrected Rand, como se vê na Tabela 4, e nas Tabelas 13 e 14. Ao realizarmos a execução do MUI em um cenário totalmente não-supervisionado, pontuando os subconjuntos de features pelo critério de maior média da silhueta entre os objetos (como não podemos calcular o NMI, o Corrected Rand e acurácia devido à ausência do label dos objetos) obtivemos a Tabela 5 para o conjunto de dados ALLAML, gerando 15 opções de seleção de features. Porém ao utilizarmos o critério da silhueta (maior valor), a seleção de features a partir do método SPEC foi escolhida, com um NMI e Corrected Rand de $0,06943$ e $0,02433$ respectivamente, mostrando-se bastante inferior ao Max, porém, entre os demais resultados, pudemos observar um NMI de $0,83644$, Corrected Rand de $0,89002$ e acurácia de $0,97222$, obtendo um desempenho superior quando comparamos com os resultados obtidos na utilização de todas as features.

Os resultados neste conjunto de dados também nos mostram uma característica que se repete nos demais conjuntos, onde na maioria dos casos, não podemos assumir que as features obtidas pela configuração selecionada através do critério da silhueta são as melhores em relação aos índices supervisionados. Por exemplo, no ALLAML, selecionando a configuração com a maior média das silhuetas, consideramos que as 6 features obtidas pelo ponto de inflexão através da execução do método SPEC, sejam as melhores, porém, esta configuração com este ponto de corte obteve um NMI igual a $0,06843$, onde existia uma outra configuração ou ponto de corte para este mesmo método capaz de alcançar um NMI de $0,45114$. Desta forma cabe ao especialista de dados escolher entre as opções geradas (as 15 possibilidades que se referem os resultados individuais mais as junções pela

contagem de Borda), qual subconjunto de features seria considerado mais consistente, fazendo-lhe mais sentido em seu modelo. Assim, podemos adicionar a análise de PCA para auxiliar o especialista para a devida tomada de decisão.

Nos experimentos, também foi possível observar que o SPEC e o iDetect aparecem, na maioria dos casos, como os melhores métodos de seleção de features (maiores valores de NMI, Corrected Rand e acurácia), quando executados de forma isolada ou pela fusão com outros métodos, inclusive entre eles. Também foi possível observar que o método LS não se destacou, através da execução isolada, em nenhuma vez, porém ele aparece muitas vezes entre os melhores resultados quando combinado com outros métodos. De fato, a combinação entre os métodos sempre aparecem com bons resultados em nossos experimentos, isso sugere que a combinação entre os métodos através da contagem de Borda, pode realmente melhorar a performance da seleção de features. A combinação com os 4 métodos também, em nenhum momento, obteve os melhores resultados, suspeitamos que ao combinar os 4 métodos podemos selecionar features indesejadas, em um trabalho futuro podemos eliminar esta possibilidade de combinação entre métodos.

Tabela 13 – Comparação do NMI ao utilizar todas as features (Ini. NMI), o máximo valor (Max NMI), maior NMI entre as opções fornecidas pelo MUI (Best MUI NMI) e o NMI obtido a partir da silhueta pelo MUI (Best Sil. NMI)

Data	Ini. NMI	Max NMI	Best MUI NMI	Best Sil. NMI
warpAR10P	0,18329	0,48040	0,43040	0,32465
warpPIE10P	0,32143	0,68259	0,42390	0,15623
pixraw10P	0,94960	0,98566	0,70702	0,70702
TOX171	0,26377	0,37611	0,15558	0,05379
ALLAML	0,17135	0,83644	0,83644	0,06843
Carcinom	0,67127	0,78350	0,67499	0,17570
SMK-CAN	0,00175	0,09836	0,05015	0,00091
ProtateGE	0,02550	0,31896	0,03603	0,03603

Ao realizar a mesma análise, feita no ALLAML, nos demais conjuntos de dados, podemos observar resultados similares. Através da Tabela 13 pudemos observar que o critério adotado da silhueta não obteve, com exceção para os datasets warpAR10P e ProstateGE, os melhores valores para o NMI, quando comparamos com o desempenho obtido ao utilizar todas as features. Porém existe, entre as opções disponibilizadas pelo MUI, uma opção capaz de melhorar o resultado para o NMI, isso ocorre quando comparamos o NMI ao utilizar todas as features (coluna Ini. NMI) com a melhor opção do MUI (coluna Best MUI NMI), exceto para os datasets pixraw10P e TOX171 que tiveram um desempenho pior ao realizar a seleção de features através da metodologia.

Em relação ao índice supervisionado Corrected Rand, entre as 15 sugestões exibidas ao final do MUI, houve melhoria para o índice em pelo menos uma delas, comparado com

Tabela 14 – Comparação do Corrected Rand ao utilizar todas as features (Ini. CR), o máximo valor (Max CR), maior Corrected Rand entre as opções fornecidas pelo MUI (Best MUI CR) e o Corrected Rand obtido a partir da silhueta pelo MUI (Best Sil. CR)

Data	Ini. CR	Max CR	Best MUI CR	Best Sil. CR
warpAR10P	0,02626	0,24917	0,21135	0,08163
warpPIE10P	0,09848	0,45756	0,14778	0,14778
pixraw10P	0,88802	0,97680	0,74709	0,74709
TOX171	0,15370	0,25572	0,10472	0,02666
ALLAML	0,23795	0,89002	0,89002	0,02433
Carcinom	0,55523	0,69022	0,58572	0,00900
SMK-CAN	-0,00121	0,11620	0,05820	-0,00199
ProtateGE	0,02276	0,33996	0,02207	0,00078

o resultado utilizando todas as features, em todos os conjuntos de dados com exceção dos datasets pixraw10P, TOX171 e ProstateGE, cf. Tabela 14. Se considerarmos apenas a solução com maior média das silhuetas, não houve melhora para o CR, quando comparado com a utilização de todas as features, exceto para o dataset warpPIE10P.

Notamos que o dataset SMK-CAN é o de maior dimensionalidade entre todos os testados, com 19993 features, e parece ser o mais difícil, com valores dos índices supervisionados baixos, bem como para a silhueta média. Podemos supor que para conjuntos de dimensionalidades tão grandes como ele ou ainda maiores, é necessária uma investigação mais aprofundada visando a melhoria dos padrões encontrados.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi proposta uma metodologia para seleção de features em um cenário não-supervisionado, capaz de combinar múltiplos métodos de seleção de features através da execução de um k-means clássico. A abordagem utilizada neste trabalho propõe disponibilizar opções de subconjuntos de features, de forma a revelar padrões relevantes nos conjuntos de dados de alta dimensionalidade, através de uma execução isolada dos métodos de seleção de features propostos na literatura, como também a partir da combinação entre eles. A combinação dos rankings gerados pelos métodos foi através da contagem de Borda que, de forma inovadora, obteve bons resultados.

Voltando às questões levantadas na introdução deste trabalho, podemos responder que os métodos de seleção de features que aqui foram utilizados não obtiveram resultados equivalentes, na maioria dos casos, e que não há um método consistentemente melhor que os demais, quando também observamos os resultados obtidos a partir da combinação entre os métodos em diferentes conjuntos de dados.

Neste trabalho foi proposto uma metodologia para selecionar, através de um critério não-supervisionada, as melhores configurações para os métodos de seleção de features, como também selecionar o subconjunto de features a partir de um número de m-melhores features. Sugerimos a utilização da maior média das silhuetas entre os objetos como critério não-supervisionado para a avaliação das features, através da execução de um k-means clássico. Apesar de termos utilizado o k-means clássico ao longo do trabalho como algoritmo de agrupamento, não há nenhuma dependência da metodologia em relação a esse algoritmo, e outros algoritmos de agrupamento podem igualmente serem utilizados dentro da metodologia proposta. Nem sempre a seleção da configuração com a maior média das silhuetas refletiu nos maiores índices supervisionados (nos nossos conjuntos de testes, classes pré-existentes eram conhecidas, permitindo computar a acurácia, NMI e Corrected Rand considerando essas classes). 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, entre as configurações finais propostas, correspondendo às melhores segundo o nosso critério não-supervisionado entre todas as configurações de cada método executado isoladamente, bem como de suas combinações utilizando a contagem de Borda, foram reveladas configurações que alcançavam bons resultados nos índices supervisionados. Ou seja, entre as configurações finais propostas temos as que possuem desempenho não muito distante dos melhores resultados possíveis, nos índices supervisionados, que na prática não podemos conhecer, pois estamos no cenário totalmente não supervisionado. Portanto, com o uso da metodologia foi possível descobrir configurações, de forma totalmente não supervisionada, que possuíam

uma concordância razoável com as classes previamente conhecidas dos conjuntos de testes. Assim podemos considerar, ao final, que a metodologia proposta (MUI) obteve resultados satisfatórios, inclusive podendo ser usada em aplicações reais. Entretanto, é possível desde já visualizar melhorias que podem ser exploradas em trabalhos futuros.

6.1 TRABALHOS FUTUROS

Acreditamos que algumas melhorias podem ser feitas no fluxo de execução da metodologia, com o potencial de obter melhores resultados. Uma dessas melhorias se refere ao critério adotado para a escolha do melhor subconjunto de features. Neste trabalho seriam realizados estudos para a definição de melhores critérios não supervisionados para a avaliação da qualidade dos clusters formados, podendo levar a consideração da combinação de diferentes critérios existentes, buscando encontrar configurações que possuem um perfil equilibrado de qualidade entre diferentes critérios, colocando este trabalho futuro numa perspectiva multi-critério.

Outras abordagens com o objetivo de melhorar a performance da metodologia são:

(T1) Utilizar técnicas não exaustivas de otimização de parâmetros;

(T2) Possibilitar a execução dos métodos de forma paralela (executando os processos por exemplo em uma GPU), de forma a obter um tempo de execução final menor. Atualmente a execução dos métodos é de forma serial, um método por vez, mas isso não significou uma inviabilidade para o uso da metodologia em ambientes reais, uma vez que o tempo de execução total foi da ordem de alguns minutos, para os conjuntos de dados utilizados;

(T3) 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;

(T4) Portar todas as implementações dos algoritmos para uma mesma linguagem ou plataforma;

(T5) O k utilizado pelo k-means se referia ao número de classes para cada conjunto de dados, porém em um cenário não-supervisionado, não temos essa informação, assim é necessário adotar estratégias para a definição deste k, inclusive o melhor k para os clusters "naturais" dos dados pode ser diferente daquele do número de classes desejado;

(T6) Aumentar o número de áreas de domínio dos conjuntos de dados;

(T7) Avaliar o desempenho da metodologia em conjunto de dados com dimensionalidades ainda superiores às utilizadas neste trabalho;

(T8) Realizar um experimento com validação manual por especialistas do domínio, 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.

REFERÊNCIAS

ABDI, H.; WILLIAMS, L. J. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, Wiley Online Library, v. 2, n. 4, p. 433–459, 2010.

ABREU, N. M. M. d.; DEL-VECCHIO, R.; VINAGRE, C.; STEVANOVIC, D. Introdução à teoria espectral de grafos com aplicações. *Notas em Matemática Aplicada*, v. 27, p. 25, 2007.

ALELYANI, S.; TANG, J.; LIU, H. Feature selection for clustering: A review. *Data Clustering: Algorithms and Applications*, v. 29, p. 110–121, 2013.

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.

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

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.

CAI, Z.; GOEBEL, R.; SALAVATIPOUR, M. R.; SHI, Y.; XU, L.; LIN, G. Selecting genes with dissimilar discrimination strength for sample class prediction. In: WORLD SCIENTIFIC. *Proceedings Of The 5th Asia-Pacific Bioinformatics Conference*. [S.l.], 2007. p. 81–90.

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

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.

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.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37, 1996.

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.

GOLUB, T. R.; SLONIM, D. K.; TAMAYO, P.; HUARD, C.; GAASENBEEK, M.; MESIROV, J. P.; COLLER, H.; LOH, M. L.; DOWNING, J. R.; CALIGIURI, M. A. et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, American Association for the Advancement of Science, v. 286, n. 5439, p. 531–537, 1999.

GU, Q.; LI, Z.; HAN, J. 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.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, v. 3, n. Mar, p. 1157–1182, 2003.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011.

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.

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

HELLER, M. J. Dna microarray technology: devices, systems, and applications. *Annual review of biomedical engineering*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 4, n. 1, p. 129–153, 2002.

HOND, D.; SPACEK, L. Distinctive descriptions for face processing. In: *BMVC*. [S.l.: s.n.], 1997. p. 0–4.

HOU, C.; NIE, F.; YI, D.; WU, Y. 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, n. 1, p. 1324.

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

KRÄHENBÜHL, P.; KOLTUN, V. Geodesic object proposals. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 725–739.

LI, J.; CHENG, K.; WANG, S.; MORSTATTER, F.; ROBERT, T.; TANG, J.; LIU, H. Feature selection: A data perspective. *arXiv:1601.07996*, 2016.

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

LIU, X.; WANG, L.; ZHANG, J.; YIN, J.; LIU, H. 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.

PILOTO, S.; SCHILLING, T. F. Ovo1 links wnt signaling with n-cadherin localization during neural crest migration. *Development*, The Company of Biologists Ltd, p. dev–048439, 2010.

RENDÓN, E.; ABUNDEZ, I.; ARIZMENDI, A.; QUIROZ, E. M. Internal versus external cluster validation indexes. *International Journal of computers and communications*, v. 5, n. 1, p. 27–34, 2011.

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.

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

SINGH, D.; FEBBO, P. G.; ROSS, K.; JACKSON, D. G.; MANOLA, J.; LADD, C.; TAMAYO, P.; RENSHAW, A. A.; D'AMICO, A. V.; RICHIE, J. P. et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer cell*, Elsevier, v. 1, n. 2, p. 203–209, 2002.

SMYTH, G.; GENTLEMAN, R.; CAREY, V.; HUBER, W.; IRIZARRY, R.; DUDOIT, S. *Statistics for Biology and Health*. [S.l.]: Springer New York:, 2005.

SOLER, J.; TENCÉ, F.; GAUBERT, L.; BUCHE, C. Data clustering and similarity. In: *FLAIRS Conference*. [S.l.: s.n.], 2013.

SPIRA, A.; BEANE, J. E.; SHAH, V.; STEILING, K.; LIU, G.; SCHEMBRI, F.; GILMAN, S.; DUMAS, Y.-M.; CALNER, P.; SEBASTIANI, P. et al. Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. *Nature medicine*, Nature Publishing Group, v. 13, n. 3, p. 361, 2007.

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.

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

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.

TUSHER, V. G.; TIBSHIRANI, R.; CHU, G. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 98, n. 9, p. 5116–5121, 2001.

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.

YANG, Y.; SHEN, H. T.; MA, Z.; HUANG, Z.; ZHOU, X. 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, n. 1, p. 1589.

YAO, J.; MAO, Q.; GOODISON, S.; MAI, V.; SUN, Y. Feature selection for unsupervised learning through local learning. *Pattern Recognition Letters*, Elsevier, v. 53, p. 100–107, 2015.

ZHANG, T.; YANG, J.; ZHAO, D.; GE, X. Linear local tangent space alignment and application to face recognition. *Neurocomputing*, Elsevier, v. 70, n. 7-9, p. 1547–1553, 2007.

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.

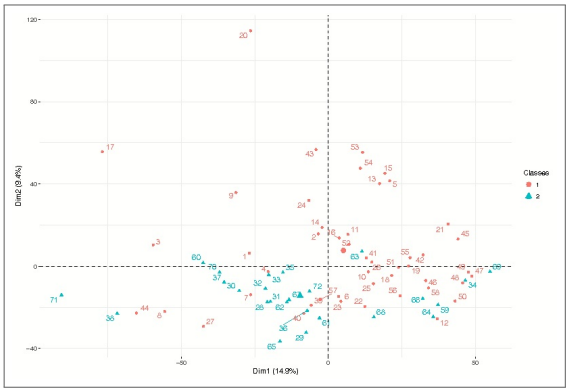
APÊNDICE A – CÓDIGO PYTHON DO MUI

Listing A.1 – Código em linguagem de programação Python do MUI

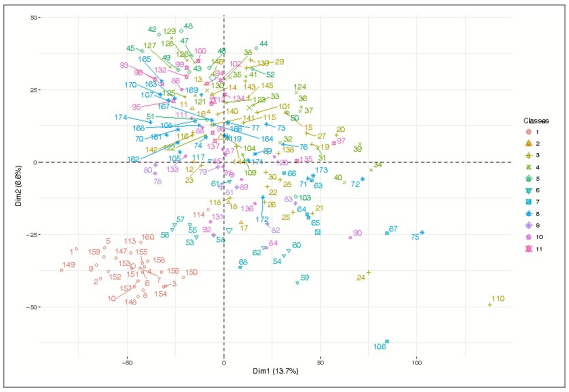
```
1
import numpy as np
3 import Chooser as cs
import scores_cutoff as cuttof
5 from utility import unsupervised_evaluation, normalize, borda_count,
    evaluate_merged_ranking
import pandas as pd
7 import scipy.io
import scipy as sp
9 from functools import partial
from itertools import repeat
11 from datetime import datetime
from execution import select_best_ranking as FSelection
13 from sklearn.feature_selection import VarianceThreshold

15 def run_parallel_FS():
    datasets = ["warpPIE10P", "pixraw10P", "warpAR10P", "TOX171", "ALLAML", "
        Carcinom", "SMK_CAN", "ProstateGE"]
17     #text datasets = ["BASEHOCK", "RELATHE", "PCMAC"]
    methods = ["LS", "SPEC", "iDetect", "GLSPFS"]
19     cols=["dataset", "total_features", "method_FS", "Method_cut", "
        selected_features", "inercia", "inercia/n_features", "nmi","accuracia",
        "corrected_rand", "f_measure"];
    dataset_folder_file_name = "train_datasets/filtered/"
21     row = []
    result = pd.DataFrame(columns=cols)
23
    for dataset_name in datasets:
25
        dataset_file_name = dataset_folder_file_name +"filtered_"+ dataset_name
            + ".csv"
27         dataset = pd.read_csv(dataset_file_name, index_col=False, sep=" ");

29         #print("rankings: "+str(rankings.shape))

31         y = list(dataset.loc[:, 'Y']);
        dataset = dataset.drop(['Y'], axis=1)
33
        data_transposed = dataset.T
35         dataset = normalize.get_normalized_data(data_transposed).T

37         #print("dataset before: "+ str(dataset.shape));

39         # eliminate features < 20% variance
        #sel = VarianceThreshold(threshold=(.1 * (1 - .1)))
41         #data = sel.fit_transform(dataset.values)

43         #if (data.shape[1] >= 10):
        #     dataset = pd.DataFrame(data);
45
```

```
        n_features = dataset.shape[1];

        rankings = np.empty([n_features, 1])

        for method in methods:

            row, best_rank = FSelection.perform_FS_parallel(dataset,
                dataset_name, method, y)
            current_row = pd.DataFrame(row, columns=cols)
            result = pd.concat([result, current_row])

            rank = pd.Series(best_rank);
            print("best rank: "+str(rank.shape))

            rankings = np.append(rankings, rank[:, None] , 1)

        np.savetxt("result_ranks_"+dataset_name+".csv", rankings, delimiter=" ")
            ;

        # LS - SPEC
        merged_ranks_LS_SPEC = borda_count.borda_sort([rankings[:,1], rankings
            [:,2]])
        borda_result_LS_SPEC = evaluate_borda(merged_ranks_LS_SPEC, dataset,
            dataset_name, 'Borda_LS_SPEC', y);
        result = pd.concat([result, borda_result_LS_SPEC])

        # LS - iDetect
        merged_ranks_LS_Idetect = borda_count.borda_sort([rankings[:,1],
            rankings[:,3]])
        borda_result_LS_Idetect = evaluate_borda(merged_ranks_LS_Idetect,
            dataset, dataset_name, 'Borda_LS_IDetect', y);
        result = pd.concat([result, borda_result_LS_Idetect])

        # LS - GLSPFS
        merged_ranks_LS_GLSPFS = borda_count.borda_sort([rankings[:,1], rankings
            [:,4]])
        borda_result_LS_GLSPFS = evaluate_borda(merged_ranks_LS_GLSPFS, dataset,
             dataset_name, 'Borda_LS_GLSPFS', y);
        result = pd.concat([result, borda_result_LS_GLSPFS])

        #SPEC - iDetect
        merged_ranks_SPEC_IDetect = borda_count.borda_sort([rankings[:,2],
            rankings[:,3]])
        borda_result_SPEC_IDetect = evaluate_borda(merged_ranks_SPEC_IDetect,
            dataset, dataset_name, 'Borda_SPEC_iDetect', y);
        result = pd.concat([result, borda_result_SPEC_IDetect])

        #SPEC - GLSPFS
        merged_ranks_SPEC_GLSPFS = borda_count.borda_sort([rankings[:,2],
            rankings[:,4]])
        borda_result_SPEC_GLSPFS = evaluate_borda(merged_ranks_SPEC_GLSPFS,
            dataset, dataset_name, 'Borda_SPEC_GLSPFS', y);
        result = pd.concat([result, borda_result_SPEC_GLSPFS])

        #iDetect - GLSPFS
        merged_ranks_iDetect_GLSPFS = borda_count.borda_sort([rankings[:,3],
            rankings[:,4]])
```

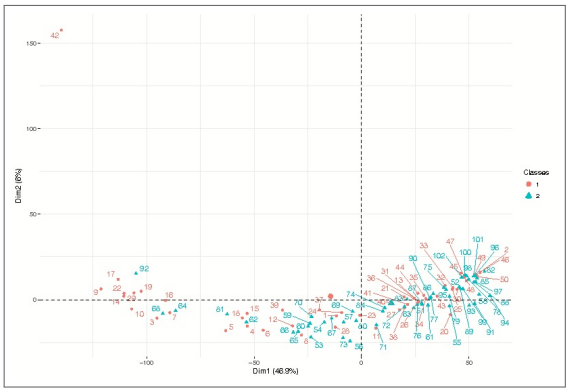
```
            borda_result_iDetect_GLSPFS = evaluate_borda(merged_ranks_iDetect_GLSPFS
                , dataset, dataset_name, 'Borda_iDetect_GLSPFS', y);
91          result = pd.concat([result, borda_result_iDetect_GLSPFS])

93          #All
            merged_ranks_all = borda_count.borda_sort([rankings[:,1], rankings[:,2],
                 rankings[:,3], rankings[:,4]])
95          borda_result_all = evaluate_borda(merged_ranks_all, dataset,
                dataset_name, 'Borda_ALL', y);
            result = pd.concat([result, borda_result_all])
97
            result.to_csv("result_all_fea_"+dataset_name+".csv", sep=" ")
99


101 def evaluate_borda(merged_ranks, dataset, dataset_name, label, y):

103     n_cluster = max(y);

105     result_borda = evaluate_merged_ranking.evaluate(merged_ranks, dataset,
            n_cluster, y, dataset_name, label)

107     cols=["dataset", "total_features", "method_FS", "Method_cut", "
            selected_features", "inercia", "inercia/n_features", "nmi","accuracia",
            "corrected_rand", "f_measure"];

109     result_tb = pd.DataFrame(result_borda, columns=cols);

111     return result_tb


113
    dt_1 = datetime.now()
115
    run_parallel_FS()
117
    dt_2 = datetime.now()
119 total_time = dt_2 - dt_1
    print("total time elapsed: "+ str(total_time))
```



(a) ALLAML

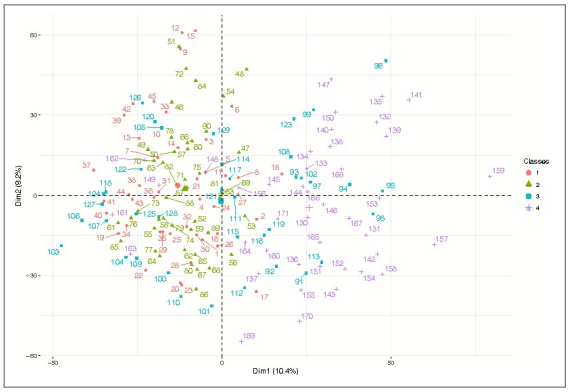


(b) Carcinom

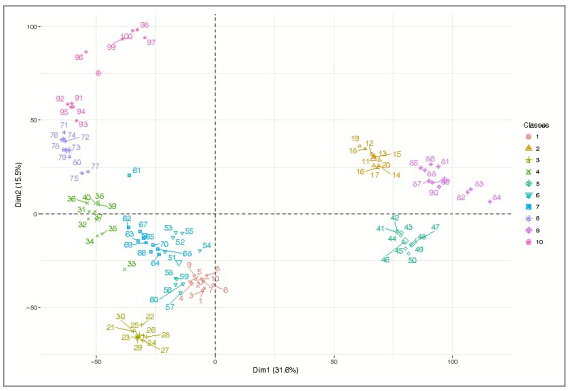


(c) ProstateGE

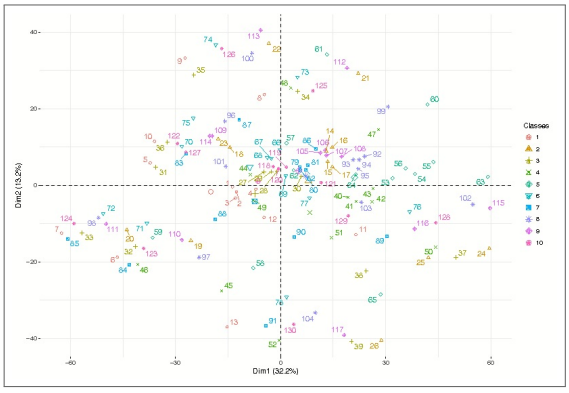
(d) SMK-CAN



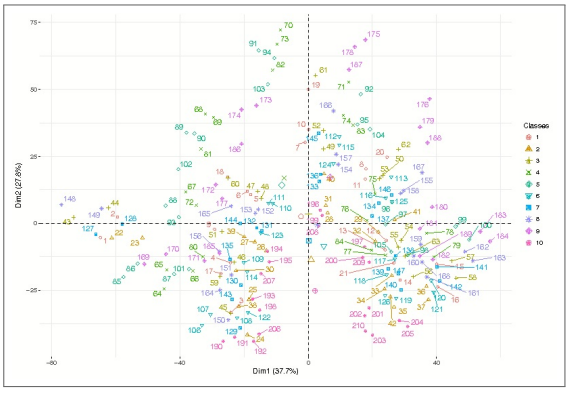
(e) TOX171



(f) pixraw10P



(g) warpAR10P



(h) warpPIE10P

Figura 24 – Análise de PCA Utilizando Todas as Features