Projeto Aprendizagem de Máquina - Turma: 2019.1

Ingryd Vanessa de Sá Teles Pereira, Levy de Souza Silva, Rafaella Leandra Souza do Nascimento

[1]Programa de Pós Graduação em Ciência da Computação - Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)

{ivstp, lss9, rlsn}@cin.ufpe.br

Resumo. *Este documento descreve o trabalho da disciplina de Aprendizagem de Máquina do semestre de 2019.1 da Universidade Federal de Pernambuco.*

1. Introdução

Este trabalho possui os objetivos de: (i) implementar, desenvolver e comparar dois classificadores supervisionados combinados. Sendo o primeiro os classificadores Bayesianos Gaussianos, e o segundo, classificadores baseados em K-Vizinhos mais próximos.

Para tal, o conjunto de dados *Multiple Features*[1] do UCI Machine Learning Repository[2] é utilizado nas avaliações experimentais. Os resultados demostram que o modelo K-Vizinhos mais próximos supera o modelo Bayesiano Gaussiano em quase 20% considerando a acurácia. Por fim, as implementações e os conjuntos de dados utlizados estão publicamente disponíveis[3].

O restante deste documento está organizado da seguinte forma: A seção 2 descreve os *datasets* utilizados nos experimentos bem como os tratamentos realizados para normalizá-los. Os modelos de aprendizagem de máquina são detalhados na Seção 3. A Seção 4 detalha a metodologia experimental. Os resultados e discussões são apontados na Seção 5 e, por fim, a Seção 6 apresenta as conclusões do trabalho.

2. Conjuntos de Dados Experimentais

Esta seção detalha o conjunto de dados utilizado nos experimentos. A Subseção 2.1 descreve os *datasets*, e a normalização aplicada nos dados é apresentada na Subseção 2.2.

2.1. *Datasets*

Neste trabalho, o conjunto de dados *Multiple Features* do *UCI Machine Learning Repository* é utilizado. Este *dataset* consiste em um conjunto de características sobre caracteres numerais manuscritos (0–9) descritas em sete arquivos. Especificamente, nós utilizamos os seguintes arquivos:

- ***mfeat-fou***: 76 características de coeficientes de *Fourier* das formas dos caracteres;
- ***mfeat-fac***: 216 atributos sobre correlação de perfil; e
- ***mfeat-kar***: 64 características de coeficientes de *Karhunen-Love*.

Cada conjunto de dados contém 2.000 instâncias com atributos do tipo reais e inteiros, as quais estão agrupadas em 10 classes. As 200 primeiras instâncias são da classe *zero*, as 200 seguintes da classe *um*, e assim sucessivamente. Os *datasets* são normalizados utilizando os critérios descritos na seguinte subseção.

[1]https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[2]https://archive.ics.uci.edu/ml/index.php
[3]https://github.com/levysouza/ProjetoAM-Cin

2.2. Normalização

A normalização dos dados é realizada utilizando a média (*u*) e o desvio padrão (*s*) das amostras. Assim, um valor *x* é normalizado para *z* conforme Equação 1.

$$z = \frac{(x - u)}{s} \tag{1}$$

3. Modelos de Aprendizagem de Máquina

Esta seção detalha os modelos de aprendizagem de máquina implementados neste trabalho. A Subseção 3.1 explica o modelo de clusterização não supervisionado MVFCMddV. O classificador combinado Bayesiano Gaussiano é detalhado na Subseção 3.2 e, por fim, a Subseção 3.3 explica o modelo combinado baseado no K-Vizinhos mais próximos.

3.1. *Multi-view relational fuzzy c-medoid vectors clustering algorithm* (MVFCMddV)

O algoritmo de clusterização *MVFCMddV* tem como objetivo agrupar elementos semelhantes considerando os seguintes diferenciais [de Carvalho et al. 2015]:

- Utilizar diferentes *views* simultaneamente;
- Representar os dados por meio de matrizes de dissimilaridade;
- Agrupar os dados utilizando uma partição *fuzzy*;
- Atribuir pesos de relevância para cada *view*;
- Descrever cada *cluster* como um vetor de *medoids*.

Assim, inicialmente, hiper-parâmetros precisam ser definidos incluindo: K que é a quantidade de *clusters*; m que ajusta a fuzzificação de cada objeto; T que representa o total máximo de iterações do modelo; e ϵ que é um valor de referência usado como critério de parada. Além disso, J define a função objetivo do algoritmo e tem os parâmetros G, Λ e U que são os vetores de *medoids*, o vetor de pesos de relevância e a distribuição *fuzzy* de cada objeto em cada *cluster* respectivamente (veja Equação 2). Durante cada iteração, um novo valor de J é calculado e o objetivo é minimizar este valor. As iterações terminam quando: (i) o número de iterações é igual a T; e (ii) a diferença entre o valor do J atual e o valor do J anterior é menor que ϵ. O algoritmo é executado como segue.

$$J(G, \Lambda, U) = \sum_{k=1}^{K} \sum_{i=1}^{n} (u_{ik})^m \sum_{j=1}^{p} \lambda_{kj} d_j(e_i g_{kj}) \tag{2}$$

Inicialização do Modelo. Três vetores G, Λ U precisam ser configurados no início do algoritmo. O vetor de G é escolhido aleatoriamente na base de dados. A dimensão deste vetor é número de *clusters* multiplicado pelo total de *views*, e cada posição possui a referência para um exemplo da base de dados que representa cada *cluster* em cada *view*. O vetor Λ possui a mesma dimensão que o G, mas cada posição armazena o peso de relevância de cada *cluster* em cada *view*. Todos os pesos são iguais a *um* inicialmente. Configurados G e Λ, o vetor U é calculado conforme Equação 3. A dimensão de U é o número de exemplos da base de dados multiplicado pelo número de *clusters*. Este vetor armazena os vetores *fuzzificados* de pertinência de cada exemplo em cada *cluster*. Finalmente, J é calculado considerando G, Λ e U por meio da Equação 2 .

$$u_{ik}^{(t)} = \left[\sum_{h=1}^{K} \left(\frac{\sum_{j=1}^{p} \lambda_{kj}^{(t)} d_j(e_i, g_{kj}^{(t)})}{\sum_{j=1}^{p} \lambda_{hj}^{(t)} d_j(e_i, g_{hj}^{(t)})} \right)^{1/(m-1)} \right]^{-1} \tag{3}$$

Execução do Modelo. Após a inicialização do modelo, a execução do algoritmo acontece em três passos como segue:

1. Primeiro, é necessário encontrar o melhor vetor de *medoids* G. Para isso, os valores de Λ e U da última iteração são fixados, e G é calculado conforme Equação 4.
2. Depois, é preciso computar o melhor vetor de pesos de relevância Λ. Para tal, o G da iteração atual e o U da iteração anterior são fixados, e o novo vetor Λ é calculado utilizando a Equação 5.
3. Finalmente, é necessário computar a melhor partição *fuzzy* U. Assim, configurados G e Λ na iteração atual, o novo U é calculado por meio da Equação 3.

$$l = \underset{1 \leq h \leq n}{argmin} \sum_{i=1}^{n} (u_{ik}^{(t-1)})^m d_j(e_i, e_h) \tag{4}$$

$$\lambda_{kj}^{(t)} = \frac{\left\{ \prod_{h=1}^{p} \left[\sum_{i=1}^{n} (u_{ik}^{(t-1)})^m d_h(e_i, g_{kh}^{(t)}) \right] \right\}^{1/p}}{\left[\sum_{i=1}^{n} (u_{ik}^{(t-1)})^m d_j(e_i, g_{kj}^{(t)}) \right]} \tag{5}$$

Finalização do Modelo. A cada iteração, novos valores G, Λ e U são obtidos, e um novo valor de J. A função objetivo do modelo é minimizar o valor de J. Assim, as iterações são repetidas até que um dos critérios de parada descritos no inicio desta seção sejam alcançados. Cada iteração produz um modelo que agrupa os dados de alguma forma. Assim, o modelo ideal é o que contém o menor valor final de J.

3.2. Classificador Bayesiano Gaussiano

Esta seção descreve o modelo Bayesiano Gaussiano implementado. Os seguintes parâmetros são computados para a construção deste classificador.

Probabilidade a Priori. Inicialmente, é necessário computar a probabilidade a priori de cada classe ($P(w_i)$). A $P(w_i)$ é calculada dividindo o total de exemplo da classe w_i pelo total de exemplos do conjunto de dados. Para computá-la, a biblioteca *GaussianNB* é utilizada por meio da função *class_prior_*.

Função Densidade. Em seguida, é adotada a distribuição normal multivariada para computar a estimativa de máxima verossimilhança de cada classe, ou seja, é considerado que a função densidade $P(x_k|w_i)$ = $P(x_k|w_i, \theta)$, com $\theta = \binom{u_i}{\Sigma}$, conforme Equações 6, 7, 8 e 9. Por fim, novamente a biblioteca *GaussianNB* é utilizada para estimar os vetores de média e desvio padrão das amostras. Para isso, as funções *bayes.theta_* e $bayes.sigma\_$ são utilizadas.

$$P(x_k|w_i, \theta) = (2\pi)^{-\frac{d}{2}} (|\Sigma^{-1}|)^{\frac{1}{2}} \exp\{-\frac{1}{2}(x_k - u_k)^{tr} \Sigma^{-1} (x_k - u_k)\} \tag{6}$$

$$u_i = \frac{1}{n}\sum_{k=1}^{n} x_k \tag{7}$$

$$\Sigma = diag(\sigma_{i1}^2, ..., \sigma_{i2}^2) \tag{8}$$

$$\sigma_{il}^2 = \frac{1}{n}\sum_{k=1}^{K}(x_{kl} - u_l)^2 \tag{9}$$

Probabilidade Posteriori. Depois de computada a função densidade, a probabilidade posteriori de cada classe é calculada utilizando a Equação 10.

$$P(w_i|x_k) = \frac{P(x_k|w_i)P(w_i)}{\sum_{r=1}^{c} P(x_k|w_r)P(w_r)} \tag{10}$$

Classificação. De posse da posteriori, a classificação de uma instância x_k acontece conforme a seguinte regra: afetar x_k a classe w_l se $P(w_l|x_k)$ = $\max_{i=1}^{10} P(w_i|x_k)$.

3.3. *K*-Vizinhos mais Próximos

Esta seção descreve o modelo baseado nos K-Vizinhos mais próximos, do inglês $KNearestNeighbor$ (KNN). Os seguintes parâmetros são computados para cada a construção deste classificador.

Distância Euclidiana. Os modelos baseados em KNN necessitam de um cálculo de distância para computar a similaridade entre duas amostras. Assim, inicialmente, é adotada a distância euclidiana conforme Equação 11 para calcular as distâncias.

$$d(x, z) = \sqrt{\sum_{i=1}^{p}(x_i - z_i)^2} \tag{11}$$

Densidade Condicional e Probabilidade a Priori. Em seguida, as Equações 12 e 13 são utilizadas para calcular a densidade condicional e a probabilidade a priori. O modelo baseado em KNN supõe que em K exemplos, com $k << n$, exitem k_j exemplos, da classe w_j. Assim, a função densidade e a probabilidade a priore são obtidas diretamente conforme as Equações 12 e 13.

$$P(x|w_j) = \frac{\frac{k_j}{n_j}}{V} \tag{12}$$

$$P(w_j) = \frac{k_j}{n_j} \tag{13}$$

Classificação. Depois, a classificação de uma instância x acontece conforme a seguinte regra, isto é, probabilidade posteriori: afetar x_k a classe w_j se:

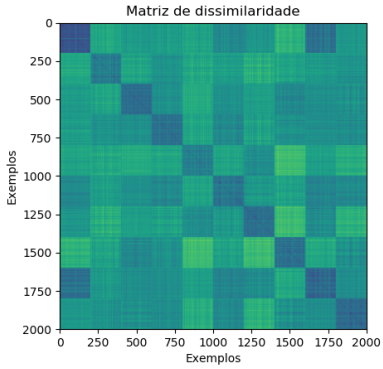
$$\frac{\frac{k_j}{n_j}}{V}\frac{k_j}{n_j} \geq \frac{\frac{k_l}{n_l}}{V}\frac{k_l}{n_l} \forall j \neq l \Rightarrow k_j \geq k_l \forall j \neq l \tag{14}$$

4. Organização dos Experimentos

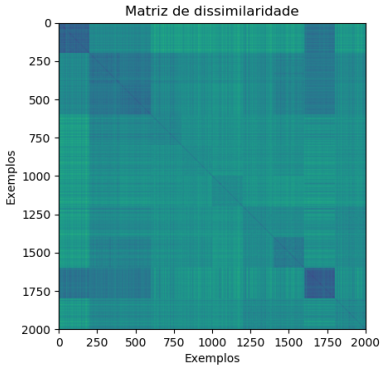
Esta seção detalha a metodologia experimental de avaliação dos modelos implementados. Primeiramente, a Subseção 4.1 apresenta as definições experimentais para o algoritmo *MVFCMddV*. E, finalmente, a metodologia de comparação dos classificadores Bayesiano Gaussiano e KNN é descrita na Subseção 4.2.

4.1. Clusterização não Supervisionada

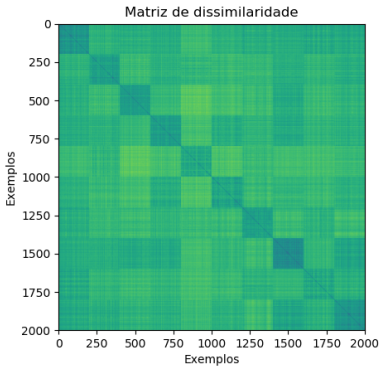
O modelo *MVFCMddV* é utilizado para a tarefa de clusterização não supervisionada. Este modelo agrupa os dados considerando múltiplas *views* ao mesmo tempo e utiliza uma matriz de dissimilaridade para cada *view* que refina o modelo. Na experimentação, três *views* de um mesmo *dataset* são utilizadas: ***mfeat-fou*** (*VIEW1*); ***mfeat-fac*** (*VIEW2*); e ***mfeat-kar*** (*VIEW3*). Além disso, os parâmetros do algoritmo são configurados como: $K = 10, m = 1.6, T = 150, \epsilon = 10^{-10}$, e 100 diferentes modelos serão criados, dos quais o melhor modelo é o que contém o menor valor de de J. Por fim, a Figura 1 apresenta uma visualização das matrizes da dissimilaridade de cada *view*.



(a) *mfeat-fou*



(b) *mfeat-fac*



(c) *mfeat-kar*

Figura 1. Matrizes de dissimilaridade de cada *view*

4.2. Modelos de classificação

Dois modelos de classificação são avaliados neste trabalho: um Bayesiano Gaussiano e um baseado em KNN. As avaliações consideram uma única configuração experimental,

ou seja, os modelos são executados no mesmo conjunto de dados de forma pareada. Os experimentos adotam uma validação cruzada estratificada "*30 times ten fold*". Assim, cada modelo é executado 30 vezes com uma validação cruzada "*ten fold*". Para cada execução, a métrica de acurácia é computada para cada modelo. Ao fim das dez execuções, a média de acurácia de cada modelo é calculada.

Além disso, cada classificador utiliza uma combinação de três modelos do mesmo tipo. No entanto, cada modelo utilizada um *dataset* diferente. Os *datasets* utilizados são: ***mfeat-fou*** (*VIEW1*); ***mfeat-fac*** (*VIEW2*); e ***mfeat-kar*** (*VIEW3*). Finalmente, os modelos são combinados utilizando a regra da soma, isto é, afetar uma instância x_k a classe w_j se o valor da Equação 15 é *máximo*. Sendo L o número de *datasets* e *P* a probabilidade posteriori de uma instância x_k em cada *dataset*.

$$(1-L)P(w_j) + P, view1(w_j|x_k) + P, view2(w_j|x_k) + P, view3(w_j|x_k) \tag{15}$$

5. Resultados e Discussões

Esta seção apresenta e discute os resultados obtidos após as analises experimentes apresentadas anteriormente. Primeiro, os resultados da tarefa de clusterização são exibidos na Subseção 5.1 e, por fim, os classificadores são comparados na Subseção 5.2.

5.1. Clusterização

Após as 100 iterações do MVFCMddV, é identificado como a melhor iteração a de número 51. Isto significa que o valor da função objetivo J é minimizado nesta iteração e então é considerada a matriz da partição *fuzzy* U para a formação da partição crisp em 10 grupos. O vetor de medoides (G) e a lista de elementos por grupo estão apresentados na Seção de Apêndices A e B respectivamente. A matriz crisp está disponível no arquivo *Matrix_Crisp*[4]. A quantidade de elementos por *cluster* é apresentada na Tabela 1.

Tabela 1. Quantidade de elementos por *Cluster*

Cluster	**Nº de elementos**
0	30
1	397
2	223
3	82
4	98
5	237
6	37
7	468
8	387
9	41
Total = 2.000	

[4]https://github.com/levysouza/ProjetoAM-Cin/tree/master/saida

Com os valores apresentados na Tabela 1, é formado um gráfico para melhor visualização das partições. Como pode ser observado, há um significativo desbalanceamento na base em relação ao número de elementos por *cluster*. Os grupos 7, 1 e 8 possuem mais de 300 objetos, enquanto os *clusters* 9, 6 e 0 estão representados por menos de 50, por exemplo. Este desbalanceamento observado é uma característica relevante, já que estes grupos serão utilizados como novos rótulos (ou classes) dos objetos das bases de dados para os algoritmos de classificação.

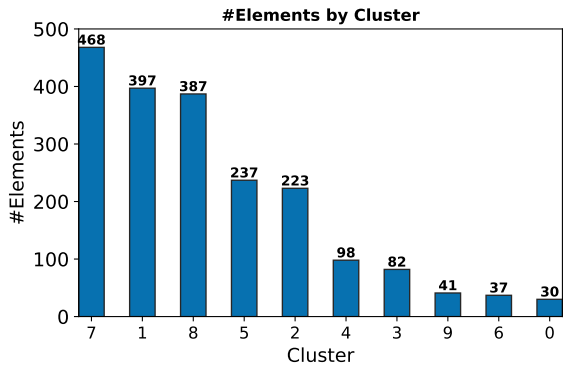


Figura 2. Total de Elementos por *Cluster*

A base de dados utilizada nos experimentos possui rótulos (informação a priori), sendo eles 0-9. A quantidade de classes rotuladas é a mesma quantidade de *clusters* do agrupamento. A Figura 3 apresenta a distribuição dos elementos por classe e por *cluster*. Na figura pode ser observado que o agrupamento não supervisionado não coincide com o agrupamento de elementos de mesma classe, ou seja, vários elementos de diferentes classes foram agrupados em um mesmo *cluster*. O agrupamento não supervisionado busca padrões desconhecidos e agrupa elementos de acordo com ele, e por essa razão os *clusters* não coincidem com as classes previamente conhecidas.

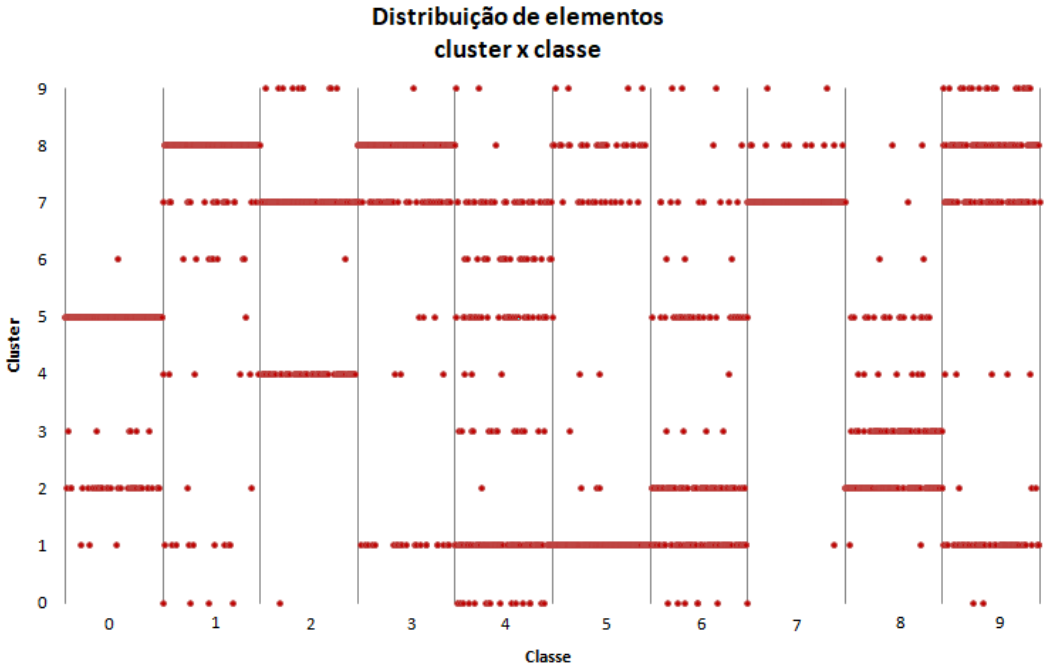


Figura 3. Elementos por *cluster* e por Classe

Validação do agrupamento: Índice Rand Ajustado. Após a determinação dos grupos pelo modelo MVFCMddV, é utilizado o Índice Rand Ajustado para validar esta tarefa [Hubert and Arabie 1985]. A validação consiste em um valor quantitativo obtido a partir do grau de correspondência entre os *clusters* formados e a informação a priori na forma de uma solução de agrupamento esperada.

O cálculo do Índice Rand Ajustado é determinado na Equação 16. Considerando n_{ij} o número de objetos que estão na mesma classe u_i (informação a priori) e *cluster* v_j, a representação $\sum_{ij}$ é a soma dos elementos que pertencem à mesma classe e ao mesmo *cluster*, $\sum_i$ é a soma dos elementos que pertencem à mesma classe, $\sum_j$ a soma dos elementos presentes no mesmo *cluster* e $\binom{n_k}{2}$ é o coeficiente binomial.

$$\frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2}\sum_j \binom{n_j}{2}\right]/\binom{n}{2}} \tag{16}$$

O índice Rand Ajustado pode apresentar valores entre -1 e 1, com 1 indicando uma partição perfeita formada pelo algoritmo de agrupamento. Para a aplicação desenvolvida nesse trabalho, foi obtido o Índice Rand Ajustado de 0.27492. Ou seja, o valor encontrado está distante do valor ótimo 1 e isso significa que as partições criadas não possuem alta correspondência com a informação a priori da base de dados. Este resultado corrobora com as informações apresentadas, e já discutidas, na Figura 3 .

5.2. Modelos de Classificação

Os modelos de classificação Bayesiano Gaussiano e KNN são utilizados nos experimentos. Porém, o modelo baseado em KNN precisa de um ajuste de granularidade, em que é

necessário definir qual o melhor valor de K para o determinado domínio. Para isso é realizada uma série de experimentos afim de definir qual o melhor modelo baseado no K em cada *view* (Subseção 5.2.1) para então realizar o comparativo entre os modelos Bayesiano Gaussiano e K-Vizinhos mais próximos (Subseção 5.2.2).

5.2.1. Ajuste de parâmetros do KNN

O modelo baseado em KNN necessita da configuração do parâmetro K que é o tamanho da janela, ou seja, o número de vizinhos considerados. Nesse contexto, esta subseção descreve a metodologia utilizada para configurar tal parâmetro. O parâmetro K é avaliado de forma isolada em cada conjunto de dados.

Inicialmente, o tamanho do parâmetro K é variado de 1 até 49 somando 2 a última janela escolhida. Depois, o modelo K-Vizinhos é executado em cada *dataset* com cada configuração de K. Para cada configuração, é executado uma validação cruzada *ten-fold* e medido a métrica de acurácia do modelo. Após isso, a média de acurácia para cada K adotado é calculada. A Figura 4(a)-(c) apresenta os resultados de cada valor do parâmetro K em cada conjunto de dados. O eixo X contém o valor de K, e o eixo Y o valor de acurácia média para cada K. Note que a escala do eixo Y está alterada para melhor distinguir a métrica de acurácia entre os valores de K.

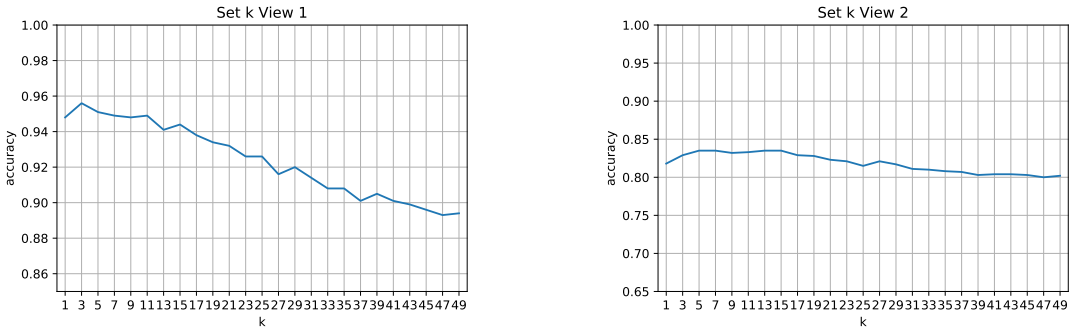
Observando o conjunto de dados *mfeat-fou*, o valor de K com melhor acurácia é 3. No *dataset mfeat-fac*, a melhor acurácia está com K nós valores 5, 7, 13 e 15. Para *mfeat-kar*, os valores de K com 5 e 7 têm melhor acurácia. Assim, nós adotamos o valor de K como 3 para o *dataset mfeat-fou*. Para os dois últimos conjuntos, existem múltiplos valores de K com melhor acurácia. Logo, nós consideramos o valor de $K = 5$ porque uma menor quantidade cálculos de distância é executada entre os vizinhos.

5.2.2. Comparativo dos algoritmos

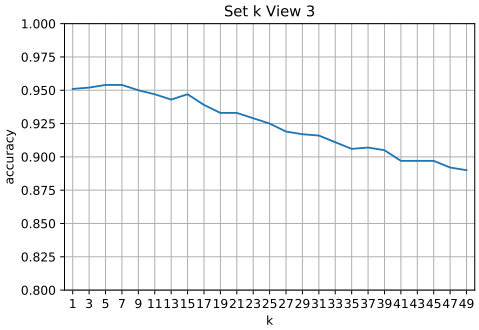
Cada experimento utiliza a validação cruzada 10-*fold* a qual é executada 30 vezes com cada algoritmo, sendo eles o Bayesiano Gaussiano e o KNN. É utilizada como medida de avaliação a taxa de acerto. Como foi gerado três modelos de cada algoritmo, um para cada *dataset*, o resultado final dos modelos em cada iteração consiste na combinação dos modelos atribuída pela regra da soma.

A Figura 5(a) possui o *boxplot* com o resultado das execuções de cada um dos experimentos. Pode-se observar que a variabilidade amostral dos modelos não é muito grande, o que indica que ambos os modelos são estáveis. Porém, pode-se ver que o modelo KNN apresenta a mediana amostral para a taxa de acerto consideravelmente superior ao do modelo Bayesiano Gaussiano.

Os experimentos foram realizados de forma pareada em cada uma das 30 execuções, nos quais os mesmos elementos da base de dados foram utilizadas no treino e no teste nos modelos Bayesiano Gaussiano e KNN. A diferença absoluta entre a acurácia de ambos os modelos está representada na Figura 5(b). Como é observado, todas a barras estão dispostas na área positiva, demostrando que o KNN obteve resultados superiores.

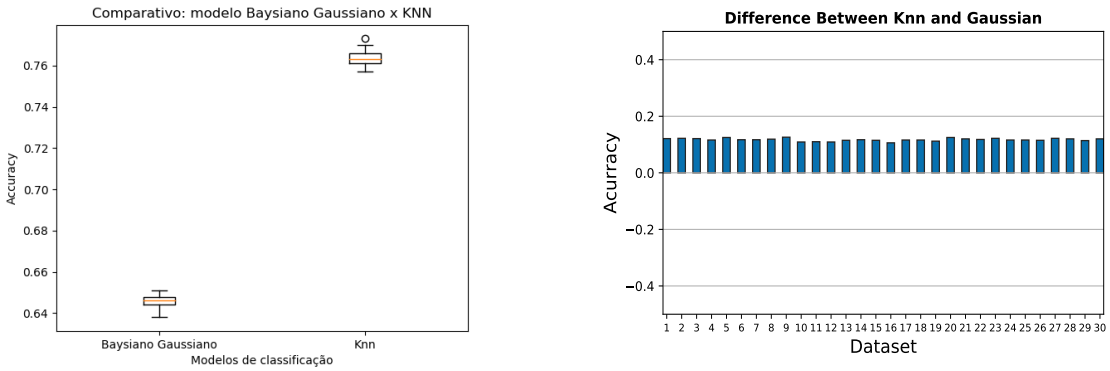


(a) *mfeat-fou* (b) *mfeat-fac*



(c) *mfeat-kar*

Figura 4. Parâmetro k do modelo KNN em cada *dataset*.



(a) (b)

Figura 5. (a) *Boxplot* com os valores das 30 execuções dos modelos Bayesiano Gaussiano e KNN; (b) Diferença absoluta entre a taxa de acerto dos modelos.

Por fim, é realizado o teste de hipóteses de Wilcoxon unilateral para amostras emparelhadas (sobre as amostras geradas nas 30 execuções de cada classificador). A hipótese nula, a de que os modelos são estatisticamente iguais, é rejeitada observando um nível de significância de 5%. O p-valor obtido no teste é de 8.906e-07, ou seja, próximo a zero o nível significância de rejeitar a hipótese nula. O intervalo de confiança de 95% para o verdadeiro valor da taxa de acerto para o classificador KNN é [0.7620; 0.7647], com

média amostral de 0.7634 e desvio padrão de 0.0036; para o Bayesiano [0.6450; 0.6472], média amostral de 0.6461 e desvio padrão de 0.0030. Portanto, em vista dos valores superiores do KNN, corroborado pelo teste de hipósteses, conclui-se que o classificador baseado em KNN fornece maior taxa de acerto em comparação ao Bayesiano Gaussiano.

Pode ser observado que os novos rótulos da base de dados apresentam grande desbalanceamento, ou seja, existem mais elementos em determinadas classes do que em outras. Isto pode impactar na precisão dos algoritmos em realizar a classificação, já que eles podem aprender mais com classes de maior frequência, e ignorar classes de menor representação acarretando baixa taxa de reconhecimento. No modelo Bayesiano Gaussiano, a decisão sobre a pertinência de um exemplo em uma classe é influenciada pela razão entre as probabilidades de ocorrência das classes, no cenário em que se tem menos representação de uma classe, a possibilidade de erro é maior. Como visto, este modelo baseado na estimativa Bayesiana obteve o pior resultado. Contudo, este desbalanceamento pode justificar o fato de os modelos não obterem taxas de acerto mais elevadas.

6. Conclusões

Este trabalho elaborou duas abordagens de Aprendizagem de Máquina: uma não-supervisionada e uma supervisionada. Assim, primeiro foi desenvolvido e avaliado o algoritmo de clusterização proposto por de Carvalho et al. (2015) e, depois, foi realizado um estudo experimental a partir da implementação de dois classificadores, um baseado em KNN e um Bayesiano Gaussiano. Os dados utilizados nos experimentos são representados por três *views*, ou seja, três conjuntos de dados disponíveis para cada elemento da base. Estas visões foram integradas para identificar a estrutura de agrupamento e classificação.

Na abordagem não-supervisionada, foi implementado o algoritmo *multi-view* MVFCMddV para encontrar grupos nos dados (i.e., *clusters*), considerando uma partição *fuzzy* gerada por meio de 3 *views*. Esta partição representa o grau de pertinência de cada elemento do conjunto de dados a cada grupo, a qual posteriormente é transformada numa partição *crisp*. Então, o maior grau obtido é o valor 1 para um determinado grupo, enquanto os demais são indicados pelo valor 0. Com isso, foi contabilizado os elementos presentes em cada grupo. Essa fase mostrou um grande desbalanceamento entre a quantidade de elementos em cada partição. Com os *clusters* formados, foi avaliado a abordagem não supervisionada a partir do Índice *Rand* Ajustado, e um valor de 0.2749 foi obtido, o que corresponde a um baixo grau de correspondência com as classes já definidas na base.

Após isso, foi iniciada a abordagem supervisionada a partir da tarefa de classificação. Os grupos formados na fase anterior corresponderam a novos rótulos para as bases (i.e., 3 *views*). Foi considerado dois classificadores para fins de comparação: um baseado em KNN e um Bayesiano Gaussiano. Como foi fornecido 3 visões dos dados, foi construído 3 modelos de cada técnica, os quais tiveram seus resultados combinados pela regra da soma. Para cada modelo foi gerado uma amostra com 30 iterações de uma validação cruzada com 10-*folds* e, com base no valor da taxa de acerto, foi avaliado e comparado o desempenho. As discussões se basearam em gráficos, *boxplots* e teste de hipóteses de Wilcoxon. Como conclusão dessa etapa foi definido o classificador baseado em K-vizinhos como o de melhor taxa de acerto para esta aplicação, o qual obteve o ganho em relação ao modelo Bayesiano Gaussiano de quase 20%.

De forma geral, o desenvolvimento deste trabalho consistiu na representação prática dos conceitos discutidos em sala de aula. Os seguintes pontos foram explorados: aprendizagem supervisionada e não supervisionada implementadas por três modelos, métricas de desempenho e avaliação de modelos de agrupamento e classificação, particionamento do conjunto de dados e combinação de modelos.

Apêndices

A. Vetor de Medoides

Tabela 2. Lista de representantes de cada grupo

Grupo	View 1	View 2	View 3
0	430	374	78
1	1025	374	78
2	1689	374	78
3	1722	374	78
4	436	374	78
5	1722	374	78
6	1722	374	78
7	436	1596	1494
8	695	374	78
9	436	374	78

B. Lista de Objetos por Grupo

- **Grupo 0**: [202, 257, 295, 344, 440, 804, 809, 817, 827, 841, 863, 869, 873, 893, 914, 925, 940, 952, 953, 978, 982, 1237, 1256, 1271, 1296, 1297, 1338, 1400, 1863, 1884]
- **Grupo 1**: [33, 50, 106, 205, 219, 228, 253, 264, 307, 327, 334, 339, 606, 615, 620, 631, 637, 673, 678, 682, 689, 690, 699, 716, 720, 730, 740, 741, 763, 765, 775, 785, 787, 800, 803, 808, 810, 811, 813, 814, 816, 821, 822, 824, 825, 830, 832, 837, 842, 845, 846, 848, 850, 852, 856, 858, 859, 864, 865, 868, 872, 876, 877, 878, 880, 881, 883, 886, 887, 888, 889, 891, 895, 896, 897, 906, 907, 912, 915, 916, 918, 919, 920, 924, 926, 929, 930, 934, 938, 944, 945, 948, 951, 954, 955, 956, 963, 964, 965, 966, 967, 968, 969, 971, 975, 977, 979, 987, 990, 992, 993, 995, 997, 999, 1002, 1003, 1005, 1006, 1008, 1009, 1010, 1011, 1012, 1013, 1016, 1017, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1033, 1035, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1055, 1057, 1061, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1080, 1082, 1083, 1084, 1085, 1086, 1088, 1089, 1093, 1094, 1096, 1099, 1102, 1103, 1105, 1106, 1108, 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1121, 1122, 1123, 1124, 1125, 1126, 1127, 1128, 1131, 1132, 1133, 1134, 1135, 1136, 1137, 1138,

1139, 1140, 1142, 1143, 1144, 1146, 1147, 1148, 1149, 1150, 1151, 1153, 1155, 1156, 1158, 1159, 1160, 1161, 1162, 1163, 1166, 1167, 1168, 1169, 1170, 1171, 1172, 1173, 1174, 1176, 1177, 1179, 1180, 1181, 1182, 1185, 1186, 1187, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1208, 1209, 1210, 1211, 1214, 1217, 1219, 1227, 1228, 1230, 1231, 1241, 1244, 1252, 1254, 1255, 1260, 1262, 1268, 1273, 1274, 1277, 1279, 1280, 1282, 1284, 1285, 1286, 1290, 1292, 1295, 1302, 1304, 1305, 1311, 1317, 1319, 1321, 1323, 1326, 1328, 1331, 1332, 1337, 1339, 1340, 1341, 1348, 1350, 1351, 1352, 1353, 1356, 1358, 1359, 1360, 1365, 1368, 1375, 1376, 1377, 1381, 1382, 1384, 1385, 1389, 1391, 1393, 1395, 1397, 1578, 1609, 1755, 1800, 1808, 1809, 1822, 1824, 1829, 1833, 1839, 1840, 1841, 1845, 1851, 1852, 1854, 1860, 1865, 1868, 1869, 1874, 1877, 1879, 1882, 1885, 1887, 1895, 1897, 1905, 1907, 1909, 1917, 1919, 1920, 1921, 1923, 1924, 1927, 1928, 1930, 1932, 1934, 1938, 1940, 1941, 1945, 1947, 1949, 1951, 1953, 1957, 1961, 1967, 1968, 1970, 1982, 1993, 1997, 1998]

- **Grupo 2**: [3, 12, 13, 36, 46, 57, 61, 66, 67, 72, 73, 76, 84, 89, 94, 108, 115, 129, 134, 136, 141, 142, 147, 152, 156, 159, 168, 170, 177, 189, 190, 193, 251, 383, 855, 1058, 1091, 1095, 1203, 1204, 1206, 1207, 1212, 1213, 1215, 1216, 1218, 1223, 1224, 1225, 1226, 1232, 1235, 1236, 1238, 1239, 1240, 1243, 1246, 1248, 1249, 1250, 1251, 1258, 1261, 1266, 1267, 1275, 1276, 1287, 1288, 1291, 1293, 1294, 1303, 1306, 1307, 1309, 1312, 1313, 1314, 1315, 1318, 1322, 1325, 1327, 1329, 1333, 1334, 1342, 1343, 1344, 1346, 1347, 1354, 1355, 1357, 1364, 1367, 1369, 1370, 1372, 1373, 1383, 1388, 1392, 1394, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1607, 1608, 1610, 1611, 1614, 1615, 1616, 1617, 1618, 1621, 1622, 1623, 1624, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1638, 1640, 1641, 1643, 1644, 1645, 1647, 1648, 1649, 1650, 1652, 1653, 1655, 1656, 1657, 1660, 1663, 1668, 1671, 1673, 1674, 1675, 1676, 1677, 1680, 1681, 1683, 1685, 1687, 1688, 1689, 1691, 1692, 1694, 1697, 1699, 1701, 1702, 1703, 1704, 1706, 1707, 1708, 1718, 1719, 1727, 1730, 1732, 1733, 1737, 1738, 1740, 1741, 1742, 1743, 1745, 1746, 1747, 1748, 1751, 1752, 1756, 1763, 1766, 1767, 1768, 1769, 1771, 1773, 1775, 1776, 1779, 1780, 1782, 1783, 1785, 1786, 1789, 1790, 1791, 1792, 1793, 1798, 1799, 1832, 1981, 1991]
- **Grupo 3**: [5, 64, 131, 135, 145, 172, 807, 812, 835, 838, 870, 875, 884, 885, 922, 927, 935, 942, 970, 981, 1036, 1234, 1269, 1316, 1349, 1613, 1620, 1626, 1627, 1637, 1651, 1654, 1658, 1661, 1662, 1664, 1665, 1666, 1670, 1672, 1679, 1684, 1686, 1693, 1695, 1698, 1700, 1709, 1710, 1711, 1713, 1715, 1716, 1717, 1721, 1722, 1723, 1724, 1725, 1726, 1728, 1731, 1734, 1735, 1744, 1749, 1753, 1760, 1762, 1764, 1770, 1774, 1777, 1778, 1781, 1784, 1787, 1788, 1794, 1795, 1796, 1797]
- **Grupo 4**: [200, 213, 266, 360, 379, 397, 404, 405, 406, 407, 411, 412, 416, 418, 419, 420, 427, 429, 431, 439, 442, 444, 451, 455, 457, 461, 469, 472, 474, 478, 482, 483, 489, 492, 496, 497, 504, 506, 507, 511, 516, 517, 518, 519, 520, 521, 522, 529, 530, 531, 533, 536, 538, 551, 557, 559, 560, 561, 564, 565, 567, 568, 569, 571, 574, 579, 580, 581, 582, 585, 586, 587, 588, 590, 591, 593, 596, 677, 687, 776, 820, 833, 894, 1056, 1097, 1362, 1625, 1639, 1667, 1705, 1736, 1750, 1758, 1805, 1827, 1901, 1933, 1980]
- **Grupo 5**: [0, 1, 2, 4, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 35, 37, 38, 39, 40, 41, 42, 43, 44, 45, 47, 48, 49,

51, 52, 53, 54, 55, 56, 58, 59, 60, 62, 63, 65, 68, 69, 70, 71, 74, 75, 77, 78, 79, 80, 81, 82, 83, 85, 86, 87, 88, 90, 91, 92, 93, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 109, 110, 111, 112, 113, 114, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 130, 132, 133, 137, 138, 139, 140, 143, 144, 146, 148, 149, 150, 151, 153, 154, 155, 157, 158, 160, 161, 162, 163, 164, 165, 166, 167, 169, 171, 173, 174, 175, 176, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 191, 192, 194, 195, 196, 197, 198, 199, 371, 726, 735, 758, 802, 815, 819, 829, 831, 834, 840, 843, 847, 853, 866, 890, 901, 908, 909, 910, 911, 917, 923, 931, 941, 946, 949, 959, 972, 980, 984, 985, 986, 1000, 1205, 1221, 1229, 1247, 1253, 1259, 1263, 1265, 1272, 1278, 1281, 1283, 1289, 1298, 1300, 1301, 1308, 1320, 1324, 1336, 1363, 1371, 1374, 1379, 1380, 1386, 1390, 1396, 1399, 1612, 1619, 1642, 1646, 1659, 1678, 1682, 1690, 1712, 1714, 1720, 1739, 1754, 1759, 1765, 1772]

- **Grupo 6**: [107, 243, 268, 296, 301, 302, 313, 365, 368, 573, 818, 826, 844, 860, 861, 867, 892, 898, 899, 902, 903, 913, 933, 937, 943, 947, 958, 961, 962, 976, 994, 998, 1233, 1270, 1366, 1669, 1761]
- **Grupo 7**: [201, 212, 215, 216, 252, 255, 258, 287, 304, 309, 312, 325, 328, 329, 333, 347, 348, 382, 391, 400, 401, 402, 403, 408, 409, 413, 414, 415, 417, 421, 422, 423, 424, 425, 426, 428, 430, 432, 433, 434, 435, 436, 438, 441, 443, 446, 447, 448, 449, 450, 452, 453, 454, 456, 458, 459, 460, 462, 463, 464, 465, 467, 468, 470, 471, 473, 475, 476, 477, 480, 481, 484, 485, 488, 490, 491, 493, 494, 495, 498, 499, 500, 501, 502, 503, 505, 508, 509, 510, 512, 513, 514, 515, 523, 524, 525, 526, 527, 528, 532, 534, 535, 537, 539, 540, 542, 543, 545, 546, 547, 548, 549, 550, 552, 553, 554, 555, 556, 562, 563, 566, 570, 572, 575, 576, 577, 578, 583, 584, 589, 592, 594, 595, 597, 598, 599, 602, 603, 610, 623, 626, 628, 633, 638, 641, 642, 643, 647, 652, 658, 659, 663, 666, 667, 671, 674, 681, 701, 706, 721, 732, 738, 743, 746, 749, 756, 760, 766, 767, 773, 781, 783, 788, 790, 805, 806, 823, 828, 836, 839, 851, 854, 857, 862, 871, 874, 879, 900, 904, 905, 921, 928, 932, 936, 939, 950, 957, 960, 973, 974, 983, 988, 989, 991, 996, 1019, 1053, 1054, 1062, 1072, 1081, 1087, 1100, 1109, 1120, 1129, 1141, 1157, 1175, 1220, 1222, 1242, 1257, 1299, 1310, 1345, 1361, 1378, 1398, 1401, 1402, 1403, 1404, 1406, 1408, 1409, 1410, 1411, 1412, 1413, 1414, 1415, 1416, 1417, 1418, 1419, 1420, 1421, 1422, 1423, 1424, 1425, 1426, 1427, 1428, 1429, 1430, 1431, 1432, 1433, 1434, 1435, 1436, 1438, 1439, 1440, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453, 1454, 1455, 1456, 1457, 1458, 1459, 1460, 1461, 1462, 1463, 1464, 1465, 1466, 1467, 1468, 1469, 1470, 1471, 1472, 1473, 1474, 1475, 1477, 1478, 1479, 1480, 1481, 1482, 1483, 1485, 1486, 1487, 1488, 1489, 1490, 1491, 1492, 1493, 1494, 1495, 1496, 1497, 1498, 1499, 1500, 1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1518, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528, 1529, 1530, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1558, 1559, 1560, 1561, 1563, 1564, 1565, 1566, 1567, 1568, 1569, 1570, 1571, 1572, 1573, 1574, 1575, 1576, 1579, 1580, 1581, 1582, 1583, 1584, 1585, 1586, 1587, 1588, 1589, 1590, 1591, 1592, 1593, 1594, 1596, 1597, 1598, 1599, 1729, 1803, 1804, 1806, 1810, 1811, 1812, 1814, 1817, 1820, 1821, 1828, 1831, 1844, 1846, 1847, 1850, 1853, 1856, 1857, 1858, 1866, 1873, 1878, 1881,

1889, 1891, 1892, 1894, 1898, 1903, 1910, 1911, 1916, 1922, 1931, 1935, 1939, 1944, 1948, 1952, 1955, 1958, 1960, 1963, 1965, 1969, 1971, 1974, 1977, 1985, 1989, 1999]

- **Grupo 8**: [3, 12, 13, 36, 46, 57, 61, 66, 67, 72, 73, 76, 84, 89, 94, 108, 115, 129, 134, 136, 141, 142, 147, 152, 156, 159, 168, 170, 177, 189, 190, 193, 251, 383, 855, 1058, 1091, 1095, 1203, 1204, 1206, 1207, 1212, 1213, 1215, 1216, 1218, 1223, 1224, 1225, 1226, 1232, 1235, 1236, 1238, 1239, 1240, 1243, 1246, 1248, 1249, 1250, 1251, 1258, 1261, 1266, 1267, 1275, 1276, 1287, 1288, 1291, 1293, 1294, 1303, 1306, 1307, 1309, 1312, 1313, 1314, 1315, 1318, 1322, 1325, 1327, 1329, 1333, 1334, 1342, 1343, 1344, 1346, 1347, 1354, 1355, 1357, 1364, 1367, 1369, 1370, 1372, 1373, 1383, 1388, 1392, 1394, 1600, 1601, 1602, 1603, 1604, 1605, 1606, 1607, 1608, 1610, 1611, 1614, 1615, 1616, 1617, 1618, 1621, 1622, 1623, 1624, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1638, 1640, 1641, 1643, 1644, 1645, 1647, 1648, 1649, 1650, 1652, 1653, 1655, 1656, 1657, 1660, 1663, 1668, 1671, 1673, 1674, 1675, 1676, 1677, 1680, 1681, 1683, 1685, 1687, 1688, 1689, 1691, 1692, 1694, 1697, 1699, 1701, 1702, 1703, 1704, 1706, 1707, 1708, 1718, 1719, 1727, 1730, 1732, 1733, 1737, 1738, 1740, 1741, 1742, 1743, 1745, 1746, 1747, 1748, 1751, 1752, 1756, 1763, 1766, 1767, 1768, 1769, 1771, 1773, 1775, 1776, 1779, 1780, 1782, 1783, 1785, 1786, 1789, 1790, 1791, 1792, 1793, 1798, 1799, 1832, 1981, 1991, 1996]
- **Grupo 9**: [410, 437, 445, 466, 479, 486, 487, 541, 544, 558, 715, 801, 849, 1007, 1032, 1154, 1184, 1245, 1264, 1335, 1441, 1562, 1801, 1813, 1837, 1843, 1855, 1861, 1875, 1888, 1893, 1902, 1904, 1908, 1950, 1956, 1964, 1966, 1973, 1975, 1979]

Referências

de Carvalho, F. d. A., de Melo, F. M., and Lechevallier, Y. (2015). A multi-view relational fuzzy c-medoid vectors clustering algorithm. *Neurocomputing*, 163:115–123.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.