Similarity Search using the NK Interaction Graph

José Carlos Bueno de Moraes, Renato Tinós

Departamento de Computação e Matemática, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP), Universidade de São Paulo (USP), Ribeirão Preto, SP, Brasil

{josecbmoraes@gmail.com ,rtinos@ffclrp.usp.br}

Abstract: *A similarity search method based on the NK interaction graph is proposed. The NK interaction graph was originally employed for clustering and is built based on distance and spatial density of the objects in a dataset. Two variations of the method are investigated. In the two variations, k objects are returned by visiting vertices of the NK interaction graph from the initial vertex related to the example of the dataset that is closer to the object to be consulted. In NK A, the k objects related to vertices with edges incident to the initial vertex are returned. In NK B, k vertices are visited starting from the initial vertex. The next visited vertex is that one with edge incident to the current vertex and that is closest to the new object to be consulted. The k objects related to the visited vertices are returned. The proposed algorithms are compared with each other and with the search for similarity based only on distance. The experimental results indicate that the proposed methods present good performance when there are clusters with arbitrary shapes in the dataset.*

Resumo: *Um método de busca por similaridade baseado no grafo de interação NK é proposto. O grafo de interação NK foi originalmente empregado para agrupamento e é construído com base na distância e densidade espacial dos objetos em um conjunto de dados. Duas variações do método são investigadas. Em NK A, os k objetos relacionados a vértices com arestas incidentes ao vértice inicial são retornados. Em NK B, k vértices são visitados a partir do vértice inicial. O próximo vértice visitado é aquele com aresta incidente ao vértice atual e que está mais próximo do novo objeto a ser consultado. Os k objetos relacionados aos vértices visitados são retornados. Os algoritmos propostos são comparados entre si e com a busca por similaridade baseada apenas na distância. Os resultados experimentais indicam que os métodos propostos apresentam bom desempenho quando existem clusters com formas arbitrárias no conjunto de dados.*

1. Introdução

Com o crescimento do volume de dados ao longo dos anos, foram desenvolvidas técnicas de busca de similaridade para responder às necessidades dos usuários em diversos segmentos do conhecimento **[**HSINCHUN *et al*., 2012**]**. A evolução das técnicas de busca de similaridade vem permitindo recuperar objetos presentes em

grandes bases de dados similares a um objeto fornecido pelo usuário de maneira eficiente, auxiliando na tomada de decisão em diversas aplicações. Por exemplo, na área da Medicina, busca por similaridade de exames (como imagens médicas, exames laboratoriais, entre outros) e laudos tem potencial para aumentar a eficiência das decisões médicas, reduzir custos e otimizar o tempo dos especialistas na análise de casos **[**CARPINETO & ROMANO, 2012**]**.

Dentre as técnicas de aprendizado de máquina supervisionada, a técnica mais comumente utilizada de busca por similaridade é aquela baseada em distância. O algoritmo dos k-vizinhos próximos (*k-nearest neighbours - KNN*) [AHA *et al*., 1991] pode ser adaptado para retornar os *k* objetos de uma base de treinamento mais similares ao objeto que está sendo consultado. No entanto, as informações sobre densidade espacial de objetos não são consideradas quando apenas *KNN* é empregado. Informações adicionais sobre densidade espacial podem ser úteis especialmente em bases de dados com agrupamentos com formas arbitrárias. A densidade espacial de objetos é explorada por algumas técnicas de clusterização para, entre outros, produzir agrupamentos que não são necessariamente hiper-esféricos [TINÓS *et al*., 2018; RODRIGUEZ & LAIO, 2014; ESTER *et al*., 1996] . Técnicas de clusterização têm sido aplicadas nas mais diversas áreas do conhecimento [HRUSCHKA *et al*., 2009]. Conceitos utilizados em clusterização podem ser especialmente úteis na recuperação de informação e na visualização de dados.

Em [TINÓS *et al*., 2018], o NKGA (*NK Hybrid Genetic Algorithm*) foi proposto para o problema de clustering. O NKGA utiliza tanto a distância entre objetos como a densidade espacial para o agrupamento de objetos. Para avaliar a soluções (particionamentos dos objetos da base de dados), o NKGA usa uma função de validação interna chamada NKCV2. Esta função utiliza informações sobre a disposição de *N* pequenos grupos de objetos, sendo *N* o número de objetos na base de dados. Cada grupo é composto de *K*+1 objetos, sendo *K* um parâmetro definido pelo usuário. As informações sobre os grupos de objetos são capturadas no *grafo de interações NK*. Tanto informações sobre densidade como de distância entre objetos são utilizadas para construir o grafo de interações NK. Resultados experimentais mostram que agrupamentos de dados com formas arbitrárias podem ser identificados usando NKGA com *K* pequeno.

Neste trabalho, propomos um método de busca por similaridade baseado no grafo de interações NK. Duas variações do método são investigadas. Nas duas variações, *k* objetos são retornados percorrendo-se o grafo de interações NK a partir do vértice inicial v_x relacionado ao objeto da base de dados mais similar ao objeto **x** a ser consultado. Na primeira variação (*método NK A*), *k*=*K*+1, sendo que os *k* objetos cujos vértices têm arestas incidentes no vértice v_x são retornados. Na segunda variação (*método NK B*), caminha-se a partir de v_x sempre alcançando o vértice, com aresta incidente, cujo objeto é mais próximo ao objeto consultado. Após *k* passos, *k* objetos são então retornados: aqueles relacionados aos *k* vértices visitados. Os algoritmos propostos são comparados entre si e com a busca por similaridade baseada em distância (chamada aqui, por simplicidade, de KNN adaptado).

2. Metodologia

Os métodos propostos são aqui comparados ao KNN adaptado, no qual, ao invés de utilizar KNN para classificar novos objetos a partir da distância para objetos de uma base de dados (também chamada de base de treinamento), utiliza-se a distância para cálculo da dissimilaridade e são retornados os k objetos mais próximos ao objeto consultado. Os métodos propostos aqui são baseados no grafo de interações NK, que é descrito a seguir.

2.1 Grafo de Interações NK

O grafo de interações NK é um grafo direcionado com N vértices, cada um com grau de entrada K+1. Dada uma base de dados (treinamento) com N objetos n-dimensionais, o primeiro passo para a construção do grafo é adicionar vértices v_i , i = 1,...,N, para cada objeto $\mathbf{y}_i$ da base de dados. Cada vértice possui auto-loop, que aqui poderia ser ignorado. Se ignorarmos os auto-loops, o grau de entrada para cada vértice é então igual a K; entretanto, para fins de uniformidade com [TINÓS *et al.*, 2018], o auto-loop será preservado na descrição do grafo, apesar de não ser utilizado pela busca por similaridade. Cada aresta (v_j , v_i) indica que o j-ésimo objeto é relacionado com o i-ésimo objeto.

É importante ressaltar que a construção das arestas leva em consideração a distância Euclidiana para objetos próximos e a densidade dos objetos. Após a criação dos vértices com auto-loop, a densidade dos objetos é calculada. Para o i-ésimo objeto, a densidade é dada por:

$$\rho_i = \sum_{j=1}^{N} \mathbf{K}(\mathbf{y}_i - \mathbf{y}_j) \tag{1}$$

sendo **K**(.) a função Kernel dada por:

$$\mathbf{K}(\mathbf{y}_i - \mathbf{y}_j) = e^{\frac{-\|\mathbf{y}_i - \mathbf{y}_j\|^2}{2\epsilon^2}} \tag{2}$$

sendo ϵ o parâmetro que define a distância de corte. Aqui, ϵ é escolhido de modo que o número médio de vizinhos de um objeto seja 2% do total de objetos da base de treinamento [RODRIGUEZ & LAIO, 2014; TINÓS *et al.,* 2018].

Para cada vértice v_i , o vértice v_{ai} representando o objeto mais próximo que possui densidade maior que o objeto relacionado a v_i é identificado. Então, uma aresta (v_{ai}, v_i) é criada. O último passo é adicionar arestas para os vértices representando objetos mais próximos até que o grau de entrada de cada vértice seja igual a K+1. A Figura 1 apresenta um exemplo do processo de criação do grafo de interações NK.

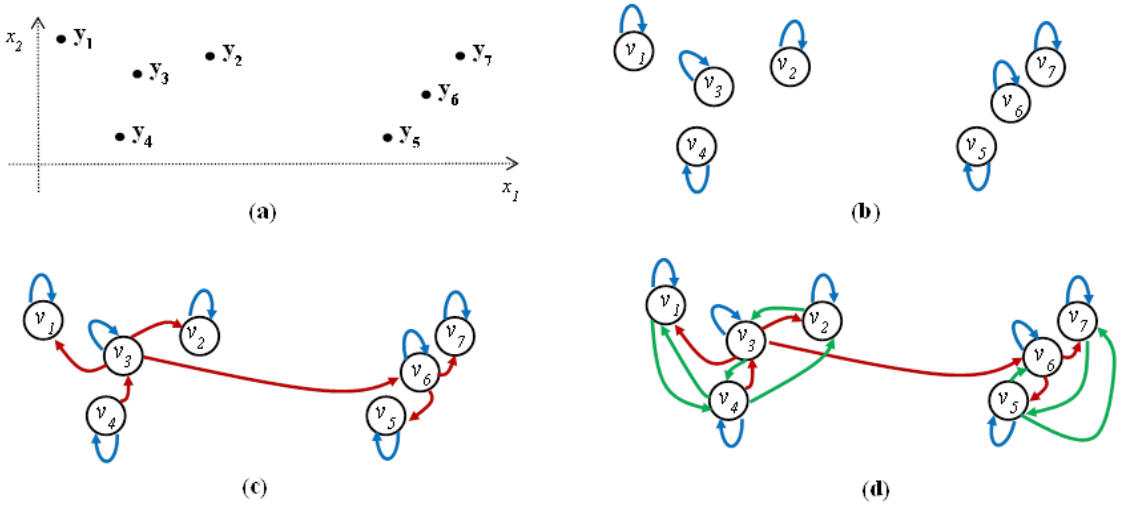

Figura 1: Exemplo de construção do grafo de interações NK com $K = 2$ para um conjunto com 7 objetos bidimensionais ($N = 7$, $n = 2$). Cada objeto da base de dados (a) é associado com um vértice com auto-loop (b). A densidade dos objetos é calculada e cada vértice é ligado ao vértice associado com o objeto mais próximo com maior densidade (c). O próximo passo é adicionar arestas para os vértices representando objetos mais próximos até que o grau de entrada de cada vértice seja igual a $K+1$. O gráfico de interações (d) tem $N = 7$ vértices e $N(K+1)$ arestas.

2.2 Busca por Similaridade via Grafo de Interações NK

Dado um novo objeto **x**, desejamos encontrar os k objetos similares a **x**. Aqui, o grafo de interações NK é utilizado para encontrar a similaridade entre objetos. O grafo de interações NK é representado por uma matriz de adjacências na qual, para cada vértice v_i , são apresentados os $(K+1)$ vértices com arestas incidentes a v_i .

Após a criação do grafo de interações NK, a próxima etapa é calcular a distância Euclidiana do novo objeto **x** para cada um dos objetos do conjunto de treinamento. O vértice relacionado ao objeto mais próximo de **x** é definido como v_x . Aqui são apresentados duas variações para a busca de similaridade baseada no grafo de interações NK. O grafo de interação é utilizado para retornar quais são os objetos mais similares ao objeto **x**. Em ambos os métodos, o grafo de interações NK é criado com $K=k-1$, sendo k o número de objetos a ser retornado pelo método. De fato, o grafo de interações NK para os dois métodos é igual, diferindo apenas a maneira como os vértices são percorridos no grafo. Vale ressaltar que, dada uma base de treinamento, o grafo de interações NK é criado uma única vez para cada valor de k.

No método *NK A*, após a identificação do vértice inicial v_x, os K nós com arestas incidentes a v_x são identificados, i.e., retorna-se a lista de adjacências para os nós incidentes a v_x. O objeto associado a v_x e os objetos associados ao $K=k-1$ vértices com arestas incidentes a v_x são então retornados pelo método como os mais similares ao objeto **x**.

No método *NK B*, após a identificação de v_x, encontra-se o vértice com arestas incidentes a v_x cujo objeto é mais próximo (de acordo com a distância Euclidiana) à **x**. Então, este novo vértice é visitado e repete-se o processo até que *k* vértices sejam visitados. Os objetos relacionados aos vértices visitados são então retornados pelo método como os mais similares ao objeto **x**.

2.3 Avaliação

Experimentos foram executados com diferentes valores de *k*. Na próxima seção, são apresentados valores de *k* entre 1 e 14, ou seja, são retornados de 1 a 14 objetos para cada objeto consultado. De modo a avaliar os métodos, cada base de dados é dividida em conjunto de treinamento e conjunto de testes. O conjunto de testes é composto pelos objetos novos que devem ser consultados em relação à similaridade para os objetos do conjunto de treinamento. Nos métodos propostos, o conjunto de treinamento é utilizado para a criação do grafo de interações NK. Na próxima seção, são realizados experimentos em que o conjunto de treinamento é composto por 90% ou 95% dos exemplos da base de dados. O restante dos dados compõe o conjunto de teste. A busca por similaridade não requer que os objetos da base de dados sejam rotulados. Entretanto, para fins de validação e comparação, consideramos que os métodos devem retornar exemplos da mesma classe que o exemplo a ser consultado. Por exemplo, em Medicina queremos que, quando uma imagem é consultada, os métodos retornem imagens da mesma classe (por exemplo, mesma doença) ou do mesmo agrupamento da imagem consultada.

Assim, para cada objeto do conjunto de teste, é calculada a acurácia em relação à classe dos objetos retornados. O valor total de acurácia para todos os objetos do conjunto de teste é então apresentado nas tabelas e figuras. A acurácia para o conjunto de teste é dada por:

$$Acc = \frac{1}{Mk} \sum_{j=1}^{M} \sum_{i=1}^{k} hit_{(j,i)} \qquad (3)$$

sendo *M* o número de objetos no conjunto de teste, *k* o número de objetos retornados pelo método e $hit_{(j,i)}$ é igual a 1 se os rótulos do *j*-ésimo objeto do conjunto de teste e do *i*-ésimo objeto retornado pelo método são iguais e 0 caso contrário.

3. Resultados

Os experimentos descritos a seguir comparam os dois modelos propostos e o KNN adaptado para três conjunto de dados do benchmark *Shape Sets* [FRÄNTI & SIERANOJA, 2018]: *Pathbased*, *Spiral* e *Aggregation.* Todos os conjuntos têm dimensão *n*=2, o que facilita a visualização da disposição dos objetos. Os dois primeiros conjuntos possuem agrupamentos que são difíceis de serem detectados por algoritmos de clustering que utilizam apenas a distância entre os objetos para o particionamento, como o *k-means*. Já o conjunto *Aggregation* possui apenas agrupamentos que são mais fáceis de serem detectados por tais algoritmos. *Pathbased* possui 300 objetos e *Spiral* possui 312 objetos, ambos com 3 clusters cada (figuras 2 e 3). Já *Aggregation* possui 788 objetos com 7 clusters. Os experimentos foram executados em um computador Intel Core i5 2,7GHz, com 8GB de memória RAM.

3.1 Base de dados Pathbased

A figura 2 mostra a disposição de dados da base *Pathbased*. Observa-se que os objetos das classes 2 e 3 formam clusters agrupados dentro de círculos, enquanto que os objetos da classe 1 apresentam dispersão ao longo de uma linha formando um semicírculo. As tabelas 1 e 2 mostram os resultados de acurácia para, respectivamente, conjuntos de testes com 5% e 10% de objetos da base de dados. São mostrados os resultados para diversos valores de *k*. Observa-se em ambas as tabelas que os métodos baseados no grafo de interações NK obtiveram melhor acurácia que o método KNN adaptado. Quando os dois métodos propostos são comparados, os melhores resultados são alcançados pelo método NK B.

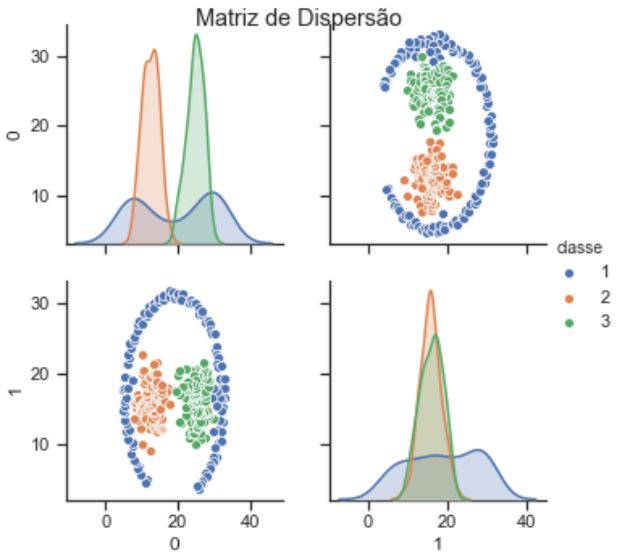

Figura 2: Matriz de dispersão do conjunto de dados *Pathbased*.

Tabela 1. Resultados para conjuntos de teste com 5% da base de dados *Pathbased*.

		k													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Acc	**KNN**	1.00	1.00	0.98	0.97	0.93	0.93	0.92	0.93	0.93	0.91	0.90	0.90	0.88	0.87
	NK A	1.00	1.00	1.00	1.00	0.99	0.97	0.96	0.95	0.96	0.95	0.95	0.93	0.93	0.92
	NK B	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.98	0.98	0.97	0.97

Tabela 2. Resultados para conjuntos de teste com 10% da base de dados *Pathbased*.

		k													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Acc	**KNN**	1.00	0.97	0.97	0.95	0.93	0.93	0.92	0.93	0.93	0.91	0.91	0.90	0.89	0.89
	NK A	1.00	1.00	1.00	0.99	0.95	0.98	0.95	0.97	0.96	0.96	0.95	0.94	0.94	0.93
	NK B	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99

3.2 Conjunto Spiral

Na Figura 3 é apresentada a matriz de dispersão do conjunto de dados *Spiral*. Observe que este conjunto é formado por objetos dispostos em 3 clusters na formas de espirais. As tabelas 3 e 4 mostram os resultados de acurácia para, respectivamente, conjuntos de testes com 5% e 10% da base de dados. Observa-se novamente em ambas as tabelas que os métodos baseados no grafo de interações NK obtiveram melhor acurácia que o método KNN adaptado. Quando os dois métodos propostos são comparados, os melhores resultados são também alcançados pelo método NK B.

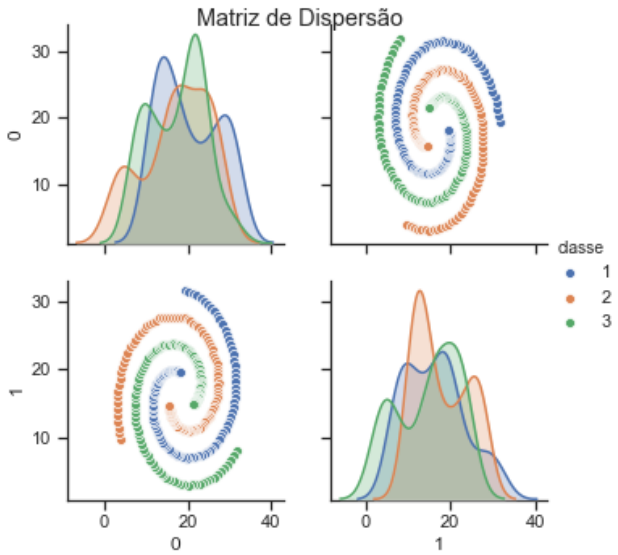

Figura 3: Matriz de dispersão do conjunto de dados *Spiral.*

Tabela 3. Resultados para conjuntos de teste com 5% da base de dados *Spiral*.

		k													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Acc	**KNN**	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.96	0.93	0.92	0.92	0.89	0.86	0.85
	NK A	1.00	1.00	1.00	1.00	1.00	1.00	0.96	0.95	0.92	0.91	0.87	0.85	0.83	0.80
	NK B	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.98	0.98	0.98

Tabela 4. Resultados para conjuntos de teste com 10% da base de dados *Spiral*.

		k													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Acc	**KNN**	1.00	1.00	1.00	1.00	0.97	0.95	0.93	0.90	0.87	0.85	0.84	0.78	0.75	0.74
	NK A	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.94	0.90	0.87	0.84	0.81	0.79
	NK B	1.00	1.00	1.00	1.00	1.00	0.98	1.00	0.98	0.98	0.97	0.97	0.96	0.95	0.94

3.3 Conjunto Aggregation

A Figura 4 apresenta a matriz de dispersão para o conjunto de dados *Aggregation*. Como pode ser observado nas tabelas 5 e 6, para esse conjunto de dados, os três métodos utilizados, tiveram uma acurácia de 100%, para o conjunto de teste com 5% dos dados. Para o conjunto de teste com 10% dos dados, apenas o método NK A apresentou acurácia abaixo de 100% (para k igual a 1 e 2). É interessante notar que nesse conjunto de dados, a matriz de dispersão indica 7 agrupamentos bem distintos, o que facilita a detecção por algoritmos de clustering que exploram a distância entre os objetos.

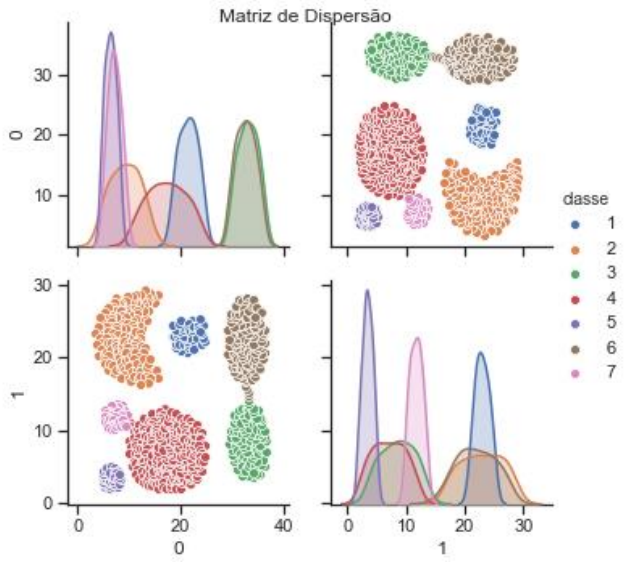

Figura 4: Matriz de dispersão do conjunto de dados *Aggregation.*

Tabela 5. Resultados para conjuntos de teste com 5% da base de dados *Aggregation*.

		k													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Acc	**KNN**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	NK A	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	NK B	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Tabela 6. Resultados para conjuntos de teste com 10% da base de dados *Aggregation*.

		k													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Acc	**KNN**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	NK A	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	NK B	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

4. Análise e Conclusões

Os resultados mostram que, ao utilizar a densidade, os métodos propostos permitiram retornar objetos dispostos nos clusters que não são necessariamente hiper-esféricos. Ao usar apenas a distância, o KNN adaptado não foi capaz de explorar a disposição de tais clusters. Os melhores resultados do KNN foram para o conjunto *Aggregation*, que possui agrupamentos compactos bem definidos. Além disso, a base de dados é maior que as outras, o que impacta a amostragem dos dados das bases de treinamento e teste.

Observa-se que, para as duas primeiras bases, conforme o número de objetos retornados (k) cresce, mais objetos de classes diferentes são retornados, ou seja, a acurácia piora. Entretanto, o impacto de k foi mais significativo para o KNN adaptado que nos métodos propostos. Em geral, diminuir o tamanho do conjunto de treinamento também implicou em diminuir a acurácia dos métodos. Como os mesmos valores de k foram testados, o número de erros aumentou para um conjunto menor de dados de treinamento. Os melhores resultados foram alcançados pelo método NK B, que percorre o grafo de interações sempre alcançando o vértice, com aresta incidente, relacionado ao objeto mais próximo do novo objeto a ser consultado. O método NK A não leva em consideração a distância para o novo objeto consultado depois que o vértice inicial é visitado. O uso da distância para o novo objeto mostrou-se útil na busca por similaridade.

Os métodos propostos se mostraram interessante para as bases de dados de formato arbitrário. Isso ocorre porque os métodos levam em consideração tanto a distância entre os objetos como também a densidade local deles. No futuro, testes com mais conjuntos de dados devem ser considerados. Além disso, testes com banco de imagens médicas devem ser realizados.

Agradecimentos

Este trabalho foi parcialmente financiado pela FAPESP (Proc. 2013/07375-0 e 2019/07665-4) e pelo CNPq (Proc. 305755/2018-8).

Referências

AHA, D. W.; KIBLER, D. & ALBERT, M.K. (1991). "Instance-based learning algorithms", *Machine Learning*, 6(1): 37-66.

CARPINETO, C. & ROMANO, G. (2012). "A survey of automatic query expansion in information retrieval", *ACM Computing Surveys (CSUR)*, 44(1).

ESTER, M.; KRIEGEL, H.-P.; SANDER, J. & XU, X. (1996), "A density-based algorithm for discovering clusters in large spatial databases with noise", *In the Proc. of the 2nd ACM Int. Conf. Knowl. Discovery Data Min. (KDD)*, 226–231.

FRÄNTI, P. & SIERANOJA, S. (2018). "K-means properties on six clustering benchmark datasets", *Applied Intelligence*, 48 (12), 4743-4759.

HRUSCHKA, E. R.; CAMPELLO, R. J. G. B.; FREITAS, A. A. & CARVALHO, A. C. P. L. F. (2009). "A survey of evolutionary algorithms for clustering", *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 39(2): 133-155.

HSINCHUN, C.; CHIANG, R. H. L. & STOREY, V. C. (2012). "Business intelligence and analytics: from big data to big impact", *MIS Quarterly,* 36(4): 1165-1188.

RODRIGUEZ, A. & LAIO, A. (2014). "Clustering by fast search and find of density peaks," *Science*, 344(6191): 1492–1496.

TINÓS, R.; ZHAO, L.; CHICANO, F. & WHITLEY, D. (2018). "NK hybrid genetic algorithm for clustering", *IEEE Transactions on Evolutionary Computation*, 22(5): 748-761.