K-Nearest Neighbors based on the Nk Interaction Graph

omitted

omitted

omitted

Abstract. *The K-Nearest Neighbors (KNN) is a simple and intuitive non-parametric classification algorithm. In KNN, the K nearest neighbors are determined according to the distance to the example to be classified. Generally, the Euclidean distance is used, which facilitates the formation of hyper-ellipsoid clusters. In this work, we propose using the Nk interaction graph to return the K-nearest neighbors in KNN. The Nk interaction graph, originally used in clustering, is built based on the distance between examples and spatial density in small groups formed by k examples of the training dataset. By using the distance combined with the spatial density, it is possible to form clusters with arbitrary shapes. We propose two variations of the KNN based on the Nk interaction graph. They differ in the way in which the vertices associated with the N examples of the training dataset are visited. The two proposed algorithms are compared to the original KNN in experiments with datasets with different properties.*

1. Introduction

Machine learning is often successfully applied in problems where discovering relationships between multiple examples in a dataset is important. If those examples carry a label, predetermining the groups they are in, the learning is called supervised [KOTSIANTIS et al., 2007]. One of the most important supervised learning problems is classification, which is a pre-requisite for different technologies that are present on our daily lives, such as speech recognition, biometrical identification, computational vision, recommendation systems and more.

Many classification algorithms are implemented using a distance metric for comparing examples, especially the Euclidean distance. This metric induces the formation of hyper ellipsoidal decision boundaries. Therefore, examples that belong to arbitrary clusters, defined by both distance and spatial density [RODRIGUEZ & LAIO, 2014], [ESTER et al., 1996] are often incorrectly classified by those algorithms.

The Euclidean distance is used the *K-Nearest Neighbors algorithm* (KNN). The KNN classifies a new example according to the majority class of the *K* nearest neighbors. The *K* nearest neighbors are determined by the Euclidean distance of the new example to the examples of the training dataset. KNN still is one of the simplest and most efficient algorithms for supervised learning, being one of the first efficient non-parametrical classification methods [FIX, 1985]. The KNN strategy is used in many other supervised and unsupervised learning algorithms.

In this work, we propose using the Nk interaction graph to find the K-nearest neighbors in KNN. We also propose two variations of the KNN based on the Nk interaction graph that differ in the way in which the vertices associated with the *N* examples of the training dataset are visited. The Nk interaction graph is built based on the distance between examples and spatial density in small groups formed by *k* examples of the training dataset. By using the distance combined with the spatial density, it is possible to form clusters with arbitrary shapes. The Nk interaction graph was initially proposed on the *NK Hybrid Genetic Algorithm* (NKGA) for clustering [TINÓS et al., 2018]. In [MORAES & TINÓS, 2020], the Nk interaction graph was used for the similarity search problem. Basically, the method proposed in [MORAES & TINÓS, 2020] returns *K* examples of a dataset similar to the queried example by visiting *K* vertices of the interaction graph. The proposed similarity search method showed to be interesting for querying examples in datasets with clusters with arbitrary shapes.

In the methods previously proposed for clustering [TINÓS et al., 2018] and similarity search [MORAES & TINÓS, 2020], for $k \geq 1$, only one edge of each vertex of the Nk Interaction Graph depends on the spatial density of the respective example of the training dataset. All the other edges depend only on the distance between examples. Here, for the modified KNN, we propose changing the ratio between the number of edges depending on spatial density and on distance on the creation of the Nk interaction graph.

2. Methodology

Both variations of the proposed KNN utilize the Nk interaction graph to build a KNN algorithm that can classify examples based on both Euclidian distance and spatial density. The original KNN and the Nk interaction graph, with modifications, are explained in sections 2.1 and 2.2. The proposed KNN based on the Nk interaction graph is presented in Section 2.3.

2.1. K-Nearest Neighbors

The KNN is a supervised learning algorithm that classifies a new example *x* based on the majority class (label) of the *K* dataset examples that are closest to *x* [AHA et al., 1991]. The *K* nearest examples are those with minimum Euclidean distance to *x*. The KNN can be modified for regression, be weighted [ALTMAN, 1992], and be modified to follow another approach of neighbor selection, where *K* is the radius of a hyper sphere that selects the examples that will be analyzed. For any version or modification, the parameter *K* has an important impact on the performance, efficiency, and definition of the decision boundaries produced by the classifier in the decision space.

2.2. Nk Interaction Graph

Most machine learning algorithms use one single metric to define the groups for clustering or decision boundaries in classification. On clustering, the Euclidean distance is often used, as in the *k-means* algorithm [MACQUEEN, 1967]. But other metrics can be employed, such as the spatial density, used in the *density-based spatial clustering of applications with noise* (DBSCAN) [ESTER et al., 1996]. The NKGA [TINÓS et al., 2018] uses both metrics, computed on examples in *N* small groups, each one with *k* examples, given a dataset with *N* examples. The groups are defined by using the Nk interaction graph.

The Nk interaction graph is a directed graph with N vertices with outdegree k. Each vertex is associated to an example of the training set. Originally, each vertex of the graph has an auto-loop, an output edge defined by spatial density and k-2 output edges defined by the Euclidean distance between examples of the training dataset. In this work, we propose to use a parameter α, which specifies the ratio of edges defined by density. Let A=$\lceil \alpha k \rceil$, where A is an integer number that represents the quantity of output edges defined by spatial density for each edge. As said before, in the original Nk graph, α always leads to A=1 and, therefore, there was one auto-loop, one density defined edge and k-2 edges defined by distance, for $k \geq 2$. For k=1, there will be only the auto-loop edge and, therefore, A=0. Here we propose using α that can result in A≠1, changing the number of edges defined by both spatial density and Euclidean distance. Each edge (v_j, v_i) of the graph indicates that the j-th example is related to the i-th example. The spatial density ρ_i for the i-th example (y_i) of the dataset with N examples is given by:

$$\rho_i = \sum_{j=i}^{N} \mathbf{K}(y_i - y_j) \tag{1}$$

where **K** is the kernel function, here defined by:

$$\mathbf{K}(y_i - y_j) = e^{\frac{-\left\|y_i - y_j\right\|^2}{2\epsilon^2}} \tag{2}$$

where ϵ is the parameter that defines the cutting distance. Here, this parameter is equal to 2%, as suggested in [RODRIGUEZ & LAIO, 2014].

For the construction of the Nk interaction graph, given a dataset with N examples, first a vertex v_i with an auto-loop is added for each example y_i of the dataset. Second, the remainder k-1 edges are defined by both spatial density and Euclidean distance, whereas its ratio is defined by an α parameter. This parameter sets the percentage of edges defined by spatial density, connecting them to the closest examples with a density greater than y_i. The number of edges connected this way is equal to A, let $A = \lceil \alpha k \rceil$, and $A \leq k$ (one of the edges must be the auto-loop). Then, remaining $k - A - 1$ edges are connected by distance, to the vertex of the closest examples to y_i. Figure 1 shows an example for building the Nk interaction graph with $k = 2$, $\alpha = 1/3$, and $N = 7$.

2.3. KNN based on the Nk Interaction Graph

The proposed KNN uses the Nk interaction graph to return the K nearest neighbors of a new example (to be classified) x. Two variations are proposed: they differ in the way the vertices associated to the N examples of the training dataset are visited. In both variations: given N examples of a training dataset and the parameters k and α, the Nk interaction graph is built (see Section 2.2); parameter k is equal to K, i.e., the outdegree for each vertex (k) is equal to the number of nearest neighbors (K) of KNN; for each new example (to be classified) x, the spatial density of x (considering all examples of the training dataset) and distances of x to all examples of the training dataset are computed; given an example x, the K visited vertices (see next paragraph) define the K nearest neighbors of x; given the K nearest neighbors of x (defined by using the Nk interaction graph), the classification is performed as in the original KNN, i.e., the KNN based on the Nk interaction graph differs from the original KNN only in the way the K nearest neighbors are defined.

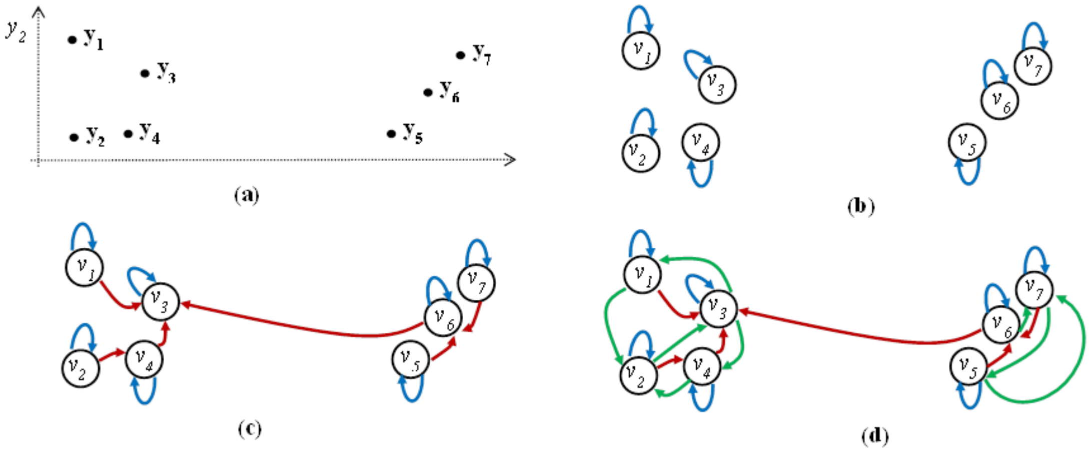


Figure 1. Building the Nk interaction graph for k=3, α=1/3, and N=7 two-dimensional examples. Each example of the dataset (a) is associated with a vertex with auto-loop (b). The density of the examples is calculated and each vertex is connected to the $A=\lceil \alpha k \rceil$ nearest examples with higher density then itself (c). Then, the remaining edges are connected to the closest vertices, resulting in the interaction graph (d) with N=7 vertices and Nk edges.

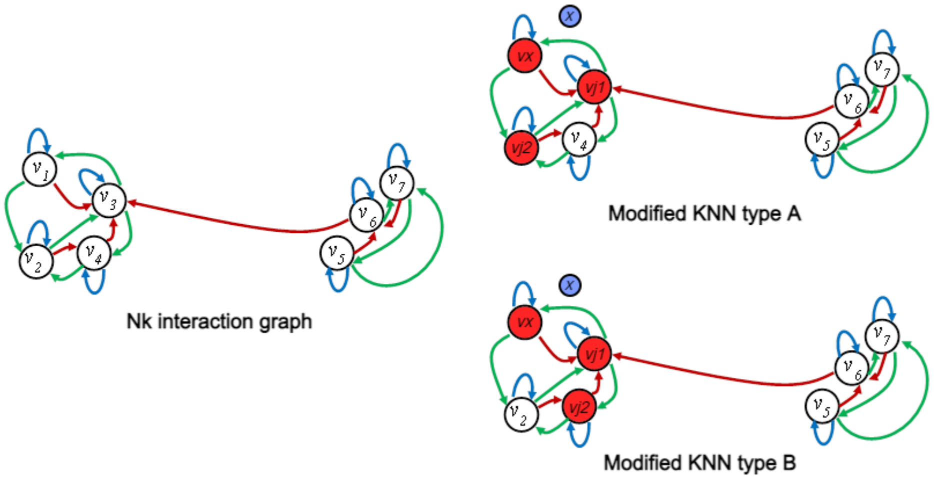
When classifying a new example x, the vertex v_x related to the example (according to the Euclidean Distance) of the training dataset that is closest to x is chosen from the training dataset to represent x on the Nk interaction graph. In the first variation of the algorithm, named *modified KNN type A*, the adjacency list of v_x is obtained by using the spatial density and distances of x to the examples of the dataset. The k=K examples associated to vertices in the adjacency list of v_x are then taken as the nearest neighbors of x and, therefore, used to classify it.

The second variation, named *modified KNN type B*, consists in finding and saving the vertex v_j, from the adjacency list of v_x, whose example y_j is the closest (according to the Euclidean distance) to x, ignoring the auto-loop. Then the operation v_x=v_j is performed, and the same step is repeated, totalizing k times. The list of vertices v_j generated through this process is taken as the nearest neighbors of x and, therefore, used to classify it. Figure 2 exemplifies the difference of modified KNN types A and B.

3. Experimental Results

The proposed algorithms are compared to the original KNN in two experiments. The experiments were designed to test the effects of changing parameters K and α. Datasets from the UCI Machine Learning Repository [DUA & GRAFF, 2019], and S-sets and Shape Datasets [FRÄNTI & SIERANOJA, 2018] are used to test the original and modified KNNs (types A and B). The 2-dimensional S-sets and Shape datasets contain clusters with different properties, e.g., clusters with non hyper-ellipsoid shapes and with different overlapping degree.

In the first experiment, the performance of the algorithms is tested for changing parameter K in the range [1, 10]. In the proposed KNN variations, we set α to a value that leads to $A=1$, i.e., only one output edge for each vertex is defined by spatial density. In the second experiment, we test the impact of changing parameter α. In this case, results were generated for two values of parameter K (5 and 10). When $K = 5$, parameter α changes resulting in A in the range [0, 4]; when $K=10$, parameter α changes resulting in A in the range [0, 9]. In the second experiment, the original KNN is tested only one time for each dataset and respective K value because it does not have α as a parameter.



Nk interaction graph

Modified KNN type A

Modified KNN type B

Figure 2. Examples of finding the K-nearest neighbors for modified KNN types A and B, when classifying an example *x*. On modified KNN type A, the neighbors are defined by the adjacency list of v_x On modified KNN type B, the neighbors are defined by the vertex adjacent to v_x related to the example that is the closest to *x*, followed by $v_j = v_x$ and iteratively repeating the process.

The results presented in the tables are for 10-fold cross-validation. Each table shows the number of test examples that are corrected classified (n) and the respective accuracy (ACC). The algorithms (modified KNN type A, modified KNN type A, and original KNN) are tested for datasets: *Aggregation*, *Compound*, *D31*, *ecoli*, *flame*, *ionosphere*, *iris*, *jain*, *pathbased*, *R15* and *spiral*; for $K = 10$ in the second experiment, we also tested the algorithms for S-sets datasets *s1*, *s2*, *s3* and *s4*.

In the tables, the results highlighted by dark gray background are the overall best results for the respective dataset, while the results highlighted in light gray background represent the best results for the respective row (for a given value of K or α, depending on the experiment).

3.1. Experiment 1: Impact of changing *K*

Table 1 shows the results for the experiment designed for testing the impact of changing parameter K. The results indicate that, in general, better performance is obtained by modified KNN type B in the UCI Machine Learning Repository datasets (*iris*, *ecoli* and *ionosphere*). These are the only datasets with dimension higher than 2. It is also possible to observe that the modified KNN type A and the original KNN presents similar results for higher values of *K*. This is explained by the small number of neighbors defined by density (*A*=1). We also observe that, in most of the datasets with hyper-ellipsoidal clusters, modified KNN type A and original KNN algorithms generally present better accuracy; the exception is for datasets with many clusters with overlap (*D31*), where better results were obtained by modified KNN type B.

3.2. Experiment 2: Impact of changing *α*

Tables 2 and 3 show the results for the experiment designed for testing the impact of changing *α*. Table 2 shows the results for *K*=5, while Table 3 shows the results for *K*=10. It is possible to observe that the modified KNN type B obtained the overall best results for datasets *D31*, *ionosphere*, and *flame* in Table 2, and in datasets *D31*, *s3* and *s4* in Table 3. The experimental results indicate that worse performance was obtained in most of the datasets for higher values of *α,* i.e., when all or most of the *K* neighbors are defined by spatial density. It is also possible to observe that better results were generally obtained for higher values of *α* on datasets with overlapping clusters (*D31, s3* and *s4*). The overlapping degree increases from *s1* to *s4* in the S-set datasets; one can observe that better results are obtained in *s3* and *s4* for higher values of *α,* i.e., choosing more neighbors by spatial density in the modified KNN impacts positively the performance when cluster overlapping degree increases. Finally, one can observe that modified KNN type A behaves like the original KNN when *A*=0.

4. Conclusions

In this work, we propose using the Nk interaction graph to find the K-nearest neighbors in KNN. We also propose changing the ratio between the number of edges depending on the spatial density and those depending only on distance in the Nk Interaction Graph. A parameter *α* controls the ratio between the number of edges depending on the spatial density and edges depending only on distance between examples.

The experimental results indicate that the best performance is generally obtained by the original KNN or the proposed KNN with a small value for *α* in datasets with 2 dimensions and non-overlapping clusters. In these cases, choosing nearest neighbors by using spatial density neutrally or negatively affects the performance of the proposed KNN. However, better results are obtained in datasets with more dimensions and/or with overlapping clusters. The results for experiments investigating the impact of *α* indicate that better results are generally obtained for small values of *α*. The exception is for datasets with overlapping clusters. Selecting more neighbors based on spatial density generally results in better performance for datasets with overlapping clusters.

The automatic selection of *α* in the proposed KNN is a possible future work. Another future work is to investigate the performance of the proposed KNN in high-dimensional datasets in Medicine. Finally, an important topic of research is to reduce the time and memory complexity of KNN. The use of the Nk interaction graph can be investigated in the future to reduce the number of examples of the training dataset analyzed by the algorithm when finding the K-nearest neighbors.

Table 1. Results for Experiment 1.

Aggregation							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	787	0.999	787	0.999	787	0.999	**788**
2	783	0.994	783	0.994	**786**	**0.997**	
3	**786**	**0.997**	785	0.996	**786**	**0.997**	
4	**784**	**0.995**	**784**	**0.995**	**784**	**0.995**	
5	**786**	**0.997**	783	0.994	**786**	**0.997**	
6	**786**	**0.997**	784	0.995	**786**	**0.997**	
7	**786**	**0.997**	784	0.995	**786**	**0.997**	
8	787	0.999	784	0.995	787	0.999	
9	**786**	**0.997**	**786**	**0.997**	**786**	**0.997**	
10	**786**	**0.997**	784	0.995	**786**	**0.997**	

Compound							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	389	0.975	389	0.975	389	0.975	**399**
2	384	0.962	384	0.962	389	0.975	
3	**382**	**0.957**	381	0.955	**382**	**0.957**	
4	**387**	**0.970**	381	0.955	**387**	**0.970**	
5	**381**	**0.955**	373	0.935	**381**	**0.955**	
6	**383**	**0.960**	372	0.932	**383**	**0.960**	
7	**379**	**0.950**	372	0.932	**379**	**0.950**	
8	**379**	**0.950**	372	0.932	**379**	**0.950**	
9	**377**	**0.945**	369	0.925	**377**	**0.945**	
10	**377**	**0.945**	369	0.925	**377**	**0.945**	

D31							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	**2981**	**0.962**	**2981**	**0.962**	**2981**	**0.962**	**3100**
2	2965	0.956	2965	0.956	**2991**	**0.965**	
3	2992	0.965	**2997**	**0.967**	2989	0.964	
4	2991	0.965	**2998**	**0.967**	2989	0.964	
5	3001	0.968	**3006**	**0.970**	3000	0.968	
6	**3004**	**0.969**	3001	0.968	**3004**	**0.969**	
7	**3000**	**0.968**	2997	0.967	**3000**	**0.968**	
8	**3000**	**0.968**	2997	0.967	**3000**	**0.968**	
9	2997	0.967	**3003**	**0.969**	2997	0.967	
10	2999	0.967	3007	0.970	2999	0.967	

ecoli							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	**274**	**0.815**	**274**	**0.815**	**274**	**0.815**	**336**
2	265	0.789	265	0.789	**274**	**0.815**	
3	285	0.848	**288**	**0.857**	285	0.848	
4	285	0.848	**287**	**0.854**	286	0.851	
5	285	0.848	283	0.842	**286**	**0.851**	
6	**286**	**0.851**	282	0.839	**286**	**0.851**	
7	**287**	**0.854**	286	0.851	**287**	**0.854**	
8	291	0.866	285	0.848	290	0.863	
9	287	0.854	287	0.854	**288**	**0.857**	
10	288	0.857	278	0.827	**289**	**0.860**	

flame							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	240	1	240	1	240	1	**240**
2	**237**	**0.988**	**237**	**0.988**	**238**	**0.992**	
3	**239**	**0.996**	238	0.992	**239**	**0.996**	
4	238	0.992	240	1.000	238	0.992	
5	239	0.996	240	1.000	239	0.996	
6	**239**	**0.996**	237	0.988	**239**	**0.996**	
7	**238**	**0.992**	236	0.983	**238**	**0.992**	
8	**238**	**0.992**	236	0.983	**238**	**0.992**	
9	**238**	**0.992**	237	0.988	**238**	**0.992**	
10	**238**	**0.992**	236	0.983	**238**	**0.992**	

ionosphere							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	306	0.872	306	0.872	306	0.872	**351**
2	310	0.883	310	0.883	307	0.875	
3	291	0.829	**295**	**0.840**	291	0.829	
4	298	0.849	**300**	**0.855**	298	0.849	
5	293	0.835	**296**	**0.843**	293	0.835	
6	**297**	**0.846**	296	0.843	**297**	**0.846**	
7	293	0.835	**298**	**0.849**	293	0.835	
8	**295**	**0.840**	292	0.832	**295**	**0.840**	
9	**293**	**0.835**	290	0.826	**293**	**0.835**	
10	**293**	**0.835**	291	0.829	**293**	**0.835**	

iris							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	**144**	**0.960**	**144**	**0.960**	**144**	**0.960**	**150**
2	139	0.927	139	0.927	**144**	**0.960**	
3	143	0.953	**144**	**0.960**	**144**	**0.960**	
4	**142**	**0.947**	**142**	**0.947**	**142**	**0.947**	
5	**143**	**0.953**	**143**	**0.953**	**143**	**0.953**	
6	142	0.947	142	0.947	**143**	**0.953**	
7	143	0.953	**144**	**0.960**	143	0.953	
8	140	0.933	**145**	**0.967**	141	0.940	
9	141	0.940	146	0.973	141	0.940	
10	142	0.947	**143**	**0.953**	**143**	**0.953**	

jain							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	373	1.000	373	1.000	373	1.000	**373**
2	370	0.992	370	0.992	373	1.000	
3	373	1.000	373	1.000	373	1.000	
4	373	1.000	373	1.000	373	1.000	
5	373	1.000	373	1.000	373	1.000	
6	373	1.000	373	1.000	373	1.000	
7	373	1.000	373	1.000	373	1.000	
8	373	1.000	373	1.000	373	1.000	
9	373	1.000	373	1.000	373	1.000	
10	373	1.000	373	1.000	373	1.000	

pathbased							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	300	1.000	300	1.000	300	1.000	**300**
2	299	0.997	299	0.997	300	1.000	
3	**298**	**0.993**	**298**	**0.993**	**298**	**0.993**	
4	**297**	**0.990**	296	0.987	**297**	**0.990**	
5	**298**	**0.993**	297	0.990	**298**	**0.993**	
6	**297**	**0.990**	**297**	**0.990**	**297**	**0.990**	
7	**297**	**0.990**	295	0.983	**297**	**0.990**	
8	296	0.987	**297**	**0.990**	296	0.987	
9	**297**	**0.990**	296	0.987	**297**	**0.990**	
10	**296**	**0.987**	295	0.983	**296**	**0.987**	

R15							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	**597**	**0.995**	**597**	**0.995**	**597**	**0.995**	**600**
2	589	0.982	589	0.982	**597**	**0.995**	
3	598	0.997	598	0.997	598	0.997	
4	**597**	**0.995**	596	0.993	**597**	**0.995**	
5	598	0.997	597	0.995	598	0.997	
6	598	0.997	596	0.993	598	0.997	
7	598	0.997	596	0.993	598	0.997	
8	598	0.997	596	0.993	598	0.997	
9	598	0.997	596	0.993	598	0.997	
10	598	0.997	596	0.993	598	0.997	

spiral							
K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
1	312	1.000	312	1.000	312	1.000	**312**
2	312	1.000	312	1.000	312	1.000	
3	312	1.000	312	1.000	312	1.000	
4	312	1.000	312	1.000	312	1.000	
5	312	1.000	312	1.000	312	1.000	
6	312	1.000	311	0.997	312	1.000	
7	312	1.000	311	0.997	312	1.000	
8	**311**	**0.997**	307	0.984	**311**	**0.997**	
9	**311**	**0.997**	304	0.974	**311**	**0.997**	
10	**308**	**0.987**	302	0.968	**308**	**0.987**	

Table 2. Results for Experiment 2 with *K*=5.

Aggregation -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	786	0.997	783	0.994	786	0.997	788
0.2 | 1	786	0.997	783	0.994			
0.4 | 2	786	0.997	783	0.994			
0.6 | 3	778	0.987	781	0.991			
0.8 | 4	773	0.981	763	0.968			

Compound -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	381	0.955	372	0.932	381	0.955	399
0.2 | 1	381	0.955	372	0.932			
0.4 | 2	381	0.955	372	0.932			
0.6 | 3	372	0.932	368	0.922			
0.8 | 4	366	0.917	344	0.862			

D31 -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	3000	0.968	3006	0.970	3000	0.968	3100
0.2 | 1	3000	0.968	3006	0.970			
0.4 | 2	3002	0.968	3007	0.970			
0.6 | 3	2960	0.955	2987	0.964			
0.8 | 4	2932	0.946	2863	0.924			

ecoli -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	286	0.851	284	0.845	286	0.851	336
0.2 | 1	286	0.851	284	0.845			
0.4 | 2	285	0.848	285	0.848			
0.6 | 3	284	0.845	281	0.836			
0.8 | 4	279	0.830	254	0.756			

flame -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	239	0.996	240	1.000	239	0.996	240
0.2 | 1	239	0.996	240	1.000			
0.4 | 2	239	0.996	240	1.000			
0.6 | 3	239	0.996	240	1.000			
0.8 | 4	238	0.992	238	0.992			

ionosphere -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	293	0.835	296	0.843	293	0.835	351
0.2 | 1	293	0.835	296	0.843			
0.4 | 2	293	0.835	297	0.846			
0.6 | 3	290	0.826	289	0.823			
0.8 | 4	281	0.801	277	0.789			

iris -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	143	0.953	143	0.953	143	0.953	150
0.2 | 1	143	0.953	143	0.953			
0.4 | 2	143	0.953	143	0.953			
0.6 | 3	141	0.940	143	0.953			
0.8 | 4	139	0.927	136	0.907			

jain -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	373	1.000	373	1.000	373	1.000	373
0.2 | 1	373	1.000	373	1.000			
0.4 | 2	373	1.000	373	1.000			
0.6 | 3	372	0.997	371	0.995			
0.8 | 4	371	0.995	365	0.979			

pathbased -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	298	0.993	297	0.990	298	0.993	300
0.2 | 1	298	0.993	297	0.990			
0.4 | 2	298	0.993	296	0.987			
0.6 | 3	296	0.987	295	0.983			
0.8 | 4	293	0.977	287	0.957			

R15 -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	598	0.997	597	0.995	598	0.997	600
0.2 | 1	598	0.997	597	0.995			
0.4 | 2	598	0.997	597	0.995			
0.6 | 3	586	0.977	588	0.980			
0.8 | 4	567	0.945	532	0.887			

spiral -> K = 5							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	312	1.000	312	1.000	312	1.000	312
0.2 | 1	312	1.000	312	1.000			
0.4 | 2	312	1.000	312	1.000			
0.6 | 3	307	0.984	311	0.997			
0.8 | 4	304	0.974	300	0.962			

Table 3. Results for Experiment 2 with *K*=10.

Aggregation -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	786	0.997	784	0.995	786	0.997	788
0.1 | 1	786	0.997	784	0.995			
0.2 | 2	786	0.997	784	0.995			
0.3 | 3	786	0.997	784	0.995			
0.4 | 4	786	0.997	784	0.995			
0.5 | 5	781	0.991	784	0.995			
0.6 | 6	775	0.984	785	0.996			
0.7 | 7	775	0.984	785	0.996			
0.8 | 8	767	0.973	774	0.982			
0.9 | 9	760	0.964	698	0.886			

Compound -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	377	0.945	369	0.925	377	0.945	399
0.1 | 1	377	0.945	369	0.925			
0.2 | 2	377	0.945	369	0.925			
0.3 | 3	377	0.945	369	0.925			
0.4 | 4	377	0.945	369	0.925			
0.5 | 5	374	0.937	369	0.925			
0.6 | 6	369	0.925	367	0.920			
0.7 | 7	369	0.925	367	0.920			
0.8 | 8	350	0.877	341	0.855			
0.9 | 9	342	0.857	277	0.694			

D31 -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	2999	0.967	3007	0.970	2999	0.967	3100
0.1 | 1	2999	0.967	3007	0.970			
0.2 | 2	2999	0.967	3007	0.970			
0.3 | 3	2999	0.967	3007	0.970			
0.4 | 4	2999	0.967	3009	0.971			
0.5 | 5	2982	0.962	3012	0.972			
0.6 | 6	2952	0.952	3014	0.972			
0.7 | 7	2952	0.952	3014	0.972			
0.8 | 8	2876	0.928	2923	0.943			
0.9 | 9	2842	0.917	2492	0.804			

ecoli -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	289	0.860	278	0.827	289	0.860	336
0.1 | 1	289	0.860	278	0.827			
0.2 | 2	289	0.860	278	0.827			
0.3 | 3	286	0.851	278	0.827			
0.4 | 4	285	0.848	278	0.827			
0.5 | 5	281	0.836	279	0.830			
0.6 | 6	282	0.839	275	0.818			
0.7 | 7	282	0.839	275	0.818			
0.8 | 8	270	0.804	264	0.786			
0.9 | 9	265	0.789	174	0.518			

Table 3. (continuation)

flame -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	238	0.992	236	0.983	238	0.992	**240**
0.1 | 1	238	0.992	236	0.983			
0.2 | 2	238	0.992	236	0.983			
0.3 | 3	238	0.992	236	0.983			
0.4 | 4	238	0.992	236	0.983			
0.5 | 5	238	0.992	236	0.983			
0.6 | 6	238	0.992	236	0.983			
0.7 | 7	238	0.992	236	0.983			
0.8 | 8	234	0.975	233	0.971			
0.9 | 9	234	0.975	217	0.904			

ionosphere -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	293	0.835	291	0.829	293	0.835	**351**
0.1 | 1	293	0.835	291	0.829			
0.2 | 2	293	0.835	291	0.829			
0.3 | 3	293	0.835	291	0.829			
0.4 | 4	293	0.835	291	0.829			
0.5 | 5	290	0.826	291	0.829			
0.6 | 6	288	0.821	290	0.826			
0.7 | 7	288	0.821	290	0.826			
0.8 | 8	281	0.801	272	0.775			
0.9 | 9	279	0.795	242	0.689			

iris -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	143	0.953	143	0.953	143	0.953	**150**
0.1 | 1	143	0.953	143	0.953			
0.2 | 2	142	0.947	142	0.947			
0.3 | 3	142	0.947	142	0.947			
0.4 | 4	141	0.940	142	0.947			
0.5 | 5	140	0.933	143	0.953			
0.6 | 6	141	0.940	142	0.947			
0.7 | 7	141	0.940	142	0.947			
0.8 | 8	136	0.907	132	0.880			
0.9 | 9	134	0.893	90	0.600			

jain -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	373	1.000	373	1.000	373	1.000	**373**
0.1 | 1	373	1.000	373	1.000			
0.2 | 2	373	1.000	373	1.000			
0.3 | 3	373	1.000	373	1.000			
0.4 | 4	373	1.000	373	1.000			
0.5 | 5	372	0.997	373	1.000			
0.6 | 6	371	0.995	372	0.997			
0.7 | 7	371	0.995	372	0.997			
0.8 | 8	365	0.979	364	0.976			
0.9 | 9	361	0.968	336	0.901			

pathbased -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	296	0.987	295	0.983	296	0.987	**300**
0.1 | 1	296	0.987	295	0.983			
0.2 | 2	296	0.987	295	0.983			
0.3 | 3	296	0.987	295	0.983			
0.4 | 4	296	0.987	295	0.983			
0.5 | 5	296	0.987	295	0.983			
0.6 | 6	295	0.983	294	0.980			
0.7 | 7	295	0.983	294	0.980			
0.8 | 8	290	0.967	280	0.933			
0.9 | 9	285	0.950	222	0.740			

R15 -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	598	0.997	596	0.993	598	0.997	**600**
0.1 | 1	598	0.997	596	0.993			
0.2 | 2	598	0.997	596	0.993			
0.3 | 3	598	0.997	596	0.993			
0.4 | 4	598	0.997	596	0.993			
0.5 | 5	593	0.988	596	0.993			
0.6 | 6	576	0.960	596	0.993			
0.7 | 7	576	0.960	596	0.993			
0.8 | 8	548	0.913	544	0.907			
0.9 | 9	531	0.885	383	0.638			

spiral -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	308	0.987	302	0.968	308	0.987	**312**
0.1 | 1	308	0.987	302	0.968			
0.2 | 2	308	0.987	302	0.968			
0.3 | 3	308	0.987	301	0.965			
0.4 | 4	308	0.987	301	0.965			
0.5 | 5	308	0.987	299	0.958			
0.6 | 6	305	0.978	297	0.952			
0.7 | 7	305	0.978	297	0.952			
0.8 | 8	296	0.949	271	0.869			
0.9 | 9	286	0.917	233	0.747			

S1 -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	2986	0.597	2973	0.595	2986	0.597	**5000**
0.1 | 1	2986	0.597	2973	0.595			
0.2 | 2	2986	0.597	2973	0.595			
0.3 | 3	2986	0.597	2973	0.595			
0.4 | 4	2986	0.597	2973	0.595			
0.5 | 5	2946	0.589	2973	0.595			
0.6 | 6	2881	0.576	2976	0.595			
0.7 | 7	2881	0.576	2976	0.595			
0.8 | 8	2823	0.565	2865	0.573			
0.9 | 9	2786	0.557	2440	0.488			

S2 -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	2843	0.569	2819	0.564	2843	0.569	**5000**
0.1 | 1	2843	0.569	2819	0.564			
0.2 | 2	2843	0.569	2819	0.564			
0.3 | 3	2843	0.569	2819	0.564			
0.4 | 4	2843	0.569	2815	0.563			
0.5 | 5	2816	0.563	2808	0.562			
0.6 | 6	2787	0.557	2808	0.562			
0.7 | 7	2787	0.557	2808	0.562			
0.8 | 8	2719	0.544	2677	0.535			
0.9 | 9	2694	0.539	2263	0.453			

S3 -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	2320	0.464	2329	0.466	2320	0.464	**5000**
0.1 | 1	2320	0.464	2329	0.466			
0.2 | 2	2318	0.464	**2327**	**0.465**			
0.3 | 3	2320	0.464	2329	0.466			
0.4 | 4	**2323**	**0.465**	2321	0.464			
0.5 | 5	2314	0.463	2318	0.464			
0.6 | 6	2302	0.460	**2328**	**0.466**			
0.7 | 7	2302	0.460	**2328**	**0.466**			
0.8 | 8	2264	0.453	2306	0.461			
0.9 | 9	2252	0.450	1968	0.394			

S4 -> K = 10							
α | α*K	A		B		KNN		Total
	n	*ACC*	*n*	*ACC*	*n*	*ACC*	
0.0 | 0	**2010**	**0.402**	2007	0.401	2010	0.402	**5000**
0.1 | 1	**2010**	**0.402**	2007	0.401			
0.2 | 2	2009	0.402	1999	0.400			
0.3 | 3	**2017**	**0.403**	1991	0.398			
0.4 | 4	**2015**	**0.403**	1985	0.397			
0.5 | 5	**2015**	**0.403**	1992	0.398			
0.6 | 6	2019	0.404	1992	0.398			
0.7 | 7	2019	0.404	1992	0.398			
0.8 | 8	1949	0.390	1954	0.391			
0.9 | 9	1939	0.388	1615	0.323			

5. Acknowledgments

Omitted.

6. References

ALTMAN, N. S. (1992). “An introduction to kernel and nearest-neighbor nonparametric regression”, The American Statistician, 46(3): 175-185.

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

DUA, D. & GRAFF, C. (2019). “UCI Machine Learning Repository”, [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science.

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.

FIX, E. (1985). “Discriminatory analysis: nonparametric discrimination, consistency properties”, Technical Report, USAF School of Aviation Medicine.

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

KOTSIANTIS, S. B.; ZAHARAKIS, I. & PINTELAS, P. (2007). “Supervised machine learning: A review of classification techniques”, Emerging artificial intelligence applications in computer engineering, 160(1): 3-24.

MACQUEEN, J. (1967) Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1, 281-297.

MORAES, J. C. B. (2020). “Busca por similaridade utilizando grafo de interações NK”, Dissertação de Mestrado em Computação Aplicada, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo.

MORAES, J. C. B., & TINÓS, R. (2020). “Busca por Similaridade usando o Grafo de Interação NK”, Nos Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional, 222-233.

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.