A Graph-based K-Nearest Neighbors Algorithm

Gustavo F. C. de Castro, 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 – Brazil

gus.castro@usp.br, rtinos@ffclrp.usp.br

Objectives

The main objective of this work is the investigation of efficient strategies using the Nk Interaction Graph [6] to return the *K* nearest examples from the training dataset that will be used to classify a new example in the K-Nearest Neighbors algorithm (KNN). The secondary objectives are: exploring the proprieties derived from the fusion of both KNN and Nk Interaction Graph; modifying the way that the Nk Interaction Graph is built; comparing the differences between the algorithms by varying separately both parameters *K*, that defines the number of edges of the graph, and *α*, that defines the ratio of edges defined by spatial density and by Euclidean distance; comparing the modified KNN variants with the original KNN algorithm using different values for the parameters, with datasets [2] [4] with different proprieties.

Materials and Methods

The proposed KNN uses the Nk Interaction Graph to return the *K* nearest neighbors of a new example (to be classified) *x*. The KNN is a simple and efficient [3] supervised learning [5] algorithm that classifies a new example *x* based on the majority class (label) of the *K* dataset examples that are closest to *x* [1]. In the standard KNN, the *K* nearest examples are those with minimum Euclidean distance to *x*.

The Nk Interaction Graph is a directed graph with *N* vertices with outdegree *k* [6]. Each vertex is associated to an example of the training set and each edge ($\boldsymbol{v}_j$, $\boldsymbol{v}_i$) of the graph indicates that the *j*-th example is related to the *i*-th example. In this work, we propose to use a parameter *α*, which specifies the ratio of edges defined by density. Let $A = \lfloor \alpha k \rfloor$ where *A* is an integer that represents the quantity of output edges defined by spatial density for each vertex. In the original Nk graph, *α* always leads to A=1 and, therefore, it had one auto-loop, one density defined edge and *k*-2 edges defined by distance, for $k \geq 2$. For $k=1$, it only had the auto-loop edge and, therefore, $A=0$. Here we propose using α that can result in $A \neq 1$, changing the number of edges defined by both spatial density and Euclidean distance.

When classifying a new example *x*, its nearest example's vertex $\boldsymbol{v}_x$ is chosen to represent *x* on the N*k* Interaction Graph. Two variations of the algorithm are proposed: in the first variation, modified KNN type A, the adjacency list of $\boldsymbol{v}_x$ is obtained by using the spatial density and distances. The $k = K$ examples in the adjacency list of $\boldsymbol{v}_x$ are then taken as the nearest neighbors of *x*; the second variation, modified KNN type B, consists in finding and saving the vertex $\boldsymbol{v}_j$, from the adjacency list of $\boldsymbol{v}_x$, whose example y_j is the closest to *x*, ignoring the auto-loop. Then the operation $\boldsymbol{v}_x = \boldsymbol{v}_j$ is performed, and the same step is repeated, totalizing *k* times. The list of vertices $\boldsymbol{v}_j$ generated through this process is taken as the nearest neighbors of *x*.

Results

The proposed algorithms are compared, regarding accuracy (ACC) and number of errors (n), to the original KNN in two experiments. The experiments were designed to test the effects of changing parameters *K* (Experiment 1) and *α* (Experiment 2), using ten-fold cross-validation. In the tables, the results highlighted by dark gray are the overall best results for the respective dataset, while the results highlighted in light gray represent the best results for the

respective row (for a given value of K or α, depending on the experiment).

Compound -> α = 0.1							
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**	
ecoli -> α = 0.1							
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**	

Table 1: Results for datasets "*ecoli*" and "*compound*" in the First Experiment.

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			

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			

Table 2: Results for datasets "*S4*" and "*pathbased*" in the Second Experiment.

Conclusions

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.

Acknowledgment: CNPq for the financial support.

References

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

[2] 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.

[3] FIX, E. (1985). "Discriminatory analysis: nonparametric discrimination, consistency properties", Technical Report, USAF School of Aviation Medicine.

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

[5] 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.

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