A Geometric Extension of KNN Classifier

Rostoum Abdelmoula Bourouba
LISIA Laboratory, Dept. of Computer Science and its Applications
Faculty of Information and Communication Technology
University Constantine 2
Constantine, Algeria
Roustoum.bourouba@univ-constantine2.dz

Khadoudja Ghanem
Dept. of Computer Science and its Applications
Faculty of Information and Communication Technology
University Constantine 2
Constantine, Algeria
Khadoudja.Ghanem@univ-constantine2.dz

Abdesslem Layeb
LISIA Laboratory, Dept. of Computer Science and its Applications
Faculty of Information and Communication Technology
University Constantine 2
Constantine, Algeria
Abdesslem.layeb@univ-constantine2.dz

***Abstract*—In this paper we present a geometric extension of K-Nearest Neighbors (KNN) classifier, one that improves classification performance by retaining the structural relationships between training samples. The proposed method is based on a geometric proximity and classification via nearest distances like KNN. Each class is represented geometrically using points and line segments. To classify a query point, it is first projected on segments, then distances from these segments are computed. Minimal distances are selected and a KNN-like voting is applied to assign the query point a label. Extensive experiments are conducted on 18 datasets and the proposed method is compared to classical kNN algorithm and eight simple and complex algorithms. Experimental results demonstrate the availability and effectiveness of the proposed algorithm.**

***Index Terms*—Classification, KNN, Geometry, Segment, Orthogonal Projection, Voronoi Diagram**

I. Introduction

Classification is among the most basic tasks of machine learning and data mining. Classification has applications in many fields such as medical diagnosis, pattern and image recognition, and anomaly detection. The K-Nearest Neighbors (KNN) classifier is one of the most well-known classification methods due to its ease of implementation, non-parametric nature, and performance in many applications. KNN uses the labels of the k closest training instances to provide a label for a test sample. The classifier uses only the distance measures in the input feature space [1]. However, the classical KNN has some significant limitations. KNN treats all training points equally informative, ignores the spatial distribution of samples within a class and is prone to noise and imbalanced classes. KNN is also usually incapable of understanding structural relationships among samples where classes have complex or non-convex boundaries.

Since its introduction by T. M. Cover and P. E. Hart in 1967 [1], various aspects of nearest neighbor methods have been explored to improve the performance of the basic KNN algorithm. These developments include algorithmic innovations, computational optimizations, and enhanced visualization techniques. Several approaches have proposed alternative distance measures [3], [4], weighted nearest neighbor techniques [2], [5], [6], [12], and fuzzy-based extensions [7]–[9].

Distance metric techniques aim to define dissimilarity measures with greater class discrimination capabilities. Weighted KNN assigns different weights either to the neighbors themselves (sample-weighted KNN) or to the features (feature-weighted KNN) to emphasize their relative importance. Fuzzy KNN, on the other hand, assigns degrees of membership to each possible class, based on both the proximity and the distribution of neighboring samples.

More recently, several optimization-based algorithms have been introduced to enhance KNN performance, including P-systems [10], Genetic Algorithms [11], and Particle Swarm Optimization (PSO) [13].

In this study, we present a new geometric approach to the KNN algorithm designed to improve accuracy by incorporating local class structures into the decision rule of the classifier. The proposed method generates intra-class line segments from training samples instead of just distances between points, to best represent the class boundaries. When classifying test points, the points are projected orthogonally onto the segments, and classification is based upon a distance-based voting rule. This modeling inherently produces a generalized Voronoi diagram, which has approval in that both points and segments ultimately influence the decision boundaries of all classes.

In addition, to improve robustness, we introduced a conflict resolution tool that eliminates ambiguous segments which are too close to points of other classes. The removal of these segments of ambiguity helps address class overlap.

The proposed method was evaluated on 18 benchmark datasets with small, high-modal, balanced and unbalanced scenarios. We institutionalized comparisons against traditional and state-of-the-art classifiers including ensemble methods. Our experimental results exhibited that the proposed method performed equal to or better than existing approaches, in terms of accuracy and generalizability, without extensive parameter

979-8-3315-6892-4/25/$31.00 © 2025 IEEE

tuning.

The remainder of the paper is organized as follows: Section II describes the proposed geometric KNN method in detail; Section III presents the experimental setup and results; Section IV discusses the findings; and Section V concludes the paper and outlines directions for future work.

II. Proposed Method

The proposed method is a geometric extension of KNN classifier. It is based on a geometric interpolation between training points of each class, creating "linear influence zones" that enrich classification compared to classical KNN. The method implicitly creates an extended Voronoi diagram where generators are both points and line segments. Each diagram corresponds to a class, the decision boundary of the class becomes more complex but flexible. These enhanced boundaries capture segments and better approximate non-convex class boundaries.

A. Segment Construction

The method starts by creating line segments between all points within the same class (Figure 1). Segments mitigate the influence of isolated points which makes the method robust against noise.

B. Conflict Segment Removal

Then a local geometry adaptation is applied to remove conflicting segments (Figure 2). Conflicting segments are segments that pass within a minimum distance of a point from another class (Eq. (1)), which allow the model to avoid ambiguities.

$$S_c = \{s \in S'_c : \forall p \in P_j,\ j \neq c,\ d(p,s) > d_{\min}(c,j)\} \quad (1)$$

$$d_{\min}(c,j) = \min(P_i - P_j) \quad (2)$$

Where: "S'_c" is the initial set of all possible segments in class c, and "$d_{\min}(c,j)$" is the minimum distance between classes c and j (Eq. (1)).

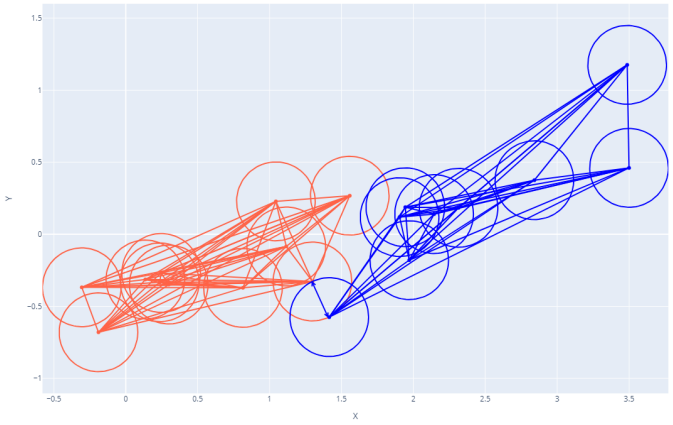

Fig. 1. Fully Connected segments.

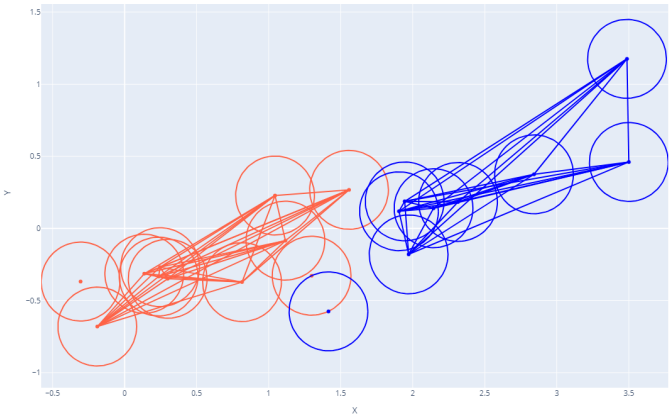

Fig. 2. Conflict-segments removed.

C. Orthogonal Projection

To classify a test point P, the algorithm first computes its orthogonal projection P' onto each valid segment $s \in S'_c$ where S'_c is the filtered set of non-conflicting segments for class c. The projection recognizes the closest point on the segment to the query point, ensuring that the distance reflects the true geometric proximity to the class structure.

The projection is determined by computing the scalar parameter t using the formula:

$$t = \frac{(P - A) \cdot (B - A)}{\|B - A\|^2} \quad (3)$$

where A and B are the endpoints of segment s. The projected point P' is then computed as follows:

- If $0 \leq t \leq 1$, the projection lies on the segment, and $P' = A + t(B - A)$,
- If $t < 0$, the projection lies before point A, and $P' = A$,
- If $t > 1$, the projection lies beyond point B, and $P' = B$.

D. Distance Calculation

After computing the orthogonal projections of the test point P onto all valid segments from each class, the algorithm calculates the Euclidean distance between P and each projected point P'. These distances represent the proximity of the test point to the geometric structure of each class:

$$d = \|P - P'\|^2 \quad (4)$$

E. Voting and Classification

A KNN-like voting mechanism is then applied. In this work, we set $k = 1$, meaning the classification is based solely on the segment closest to the test point. The fact that k equals 1 makes the method parameter-free.

The predicted class label is defined by the following formula:

$$\hat{c}(P) = \arg\min_{c \in C} \left\{ \min_{s \in S'_c} d(P,s)^2 \right\} \quad (5)$$

Where:

- C is the set of all classes,
- S'_c is the set of valid (non-conflicting) line segments for class c,

- $d(P, s)$ is the Euclidean distance between the test point P and segment s.

This approach enables the classifier to rely on spatial relationships between class members, resulting in more robust and accurate predictions compared to traditional KNN, which relies only on distances to isolated points.

Algorithm 1 Pseudo Code of Geometric Extension of KNN

```
Create line segments between all points within the same class;
Remove conflicting segments using Equation (1) and Equation (2);
for each testing point P do
    Compute its orthogonal projection P' onto all line segments (see Algorithm 2);
    Compute the perpendicular distances between P and each segment using Equation (4);
    Classify P based on the global minimum distance via voting using Equation (5);
end for
```

Algorithm 2 Orthogonal Projection of a Point onto a Segment (Figure 3)

```
Let A and B be two endpoints of the same-class segment, and P be the testing point;
Compute the scalar parameter t using Equation (3);
if 0 ≤ t ≤ 1 then
    The projection P' lies on the segment: P' = A + t(B − A);
else if t < 0 then
    The projection lies before point A: P' = A;
else
    The projection lies beyond point B: P' = B;
end if
```

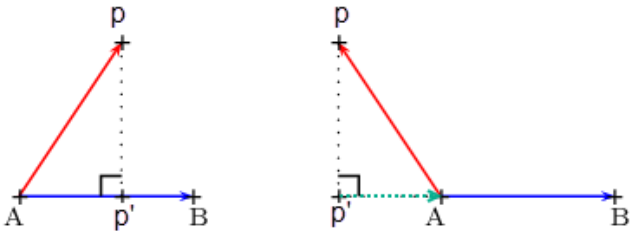

Fig. 3. Orthogonal Projection.

III. Experiments and Results

In our experiments we developed different variants of the proposed method, indeed in order to enhance its performances we have explored the combination of the proposed method with different sample selection methods and three of the tested ones which are CBD, Gabriel Graph Reduction and Tomek links are compared with the basic method. As a result best results are obtained by the basic method without any sample selection (Table II). After that we compared the basic method to 9 other classifiers, namely: KNN, Logistic Regression, Decision Tree, SVM, Neural Network, Naive Bayes, Adaboost, Gradient Boost and Random Forest. Obtained results are presented in Table III.

A. Settings

Our code is executed on a machine equipped with an Intel Core i3-14100 processor (4 cores, 3.5 GHz) and 16 GB RAM. All experiments were performed using 5-fold cross-validation, and each test was repeated 10 times to reduce randomness. The average results across repetitions were reported. All the parameter settings of main algorithms used in this experiment are the default settings.

All performance measures used in this study are the following:

a. **Accuracy**:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Where: TP (True Positives), TN (True Negatives), FP (False Positives), FN (False Negatives).

b. In addition, the non-parametric statistical test of Friedman is utilized to statistically compare obtained results.

B. Datasets Description

Performance evaluation was carried out on a total of 18 small, large and high-dimensional (with a significant number of features), balanced and unbalanced datasets obtained from the UCI Machine Learning Repository: https://archive.ics.uci.edu/, as described in Table I.

A preprocessing step precedes any use of dataset. This preprocessing consists of handling missing values, removing inconsistencies, and performing initial data cleaning to ensure the dataset is complete and suitable for modeling.

TABLE I
Details of datasets used in our experimental studies

No.	Dataset Name	Instances	Features	Classes
01	Contact Lens	24	4	3
02	Placement	215	15	2
03	Breast Cancer (Wisconsin Diagnostic)	569	30	2
04	Sonar (Mines vs Rocks)	208	60	2
05	Digits	1797	64	10
06	German Credit	1000	20	2
07	Glass Identification	214	9	6
08	Heart Disease (Cleveland)	303	13	2
09	Ionosphere	351	34	2
10	Iris	150	4	3
11	Lung Cancer	32	56	3
12	Lymphography	148	18	4
13	Mushroom	8124	22	2
14	SPECT Heart	267	22	2
15	Tic-Tac-Toe Endgame	958	9	2
16	Titanic	891	12	2
17	Wine	178	13	3
18	Zoo	101	16	7

TABLE II
OBTAINED ACCURACIES OF ALL PROPOSED VARIANTS

Datasets	CBD	Gabriel Graph Reduction	Tomek Links	Basic Model
ContactLens	0.59	0.69	0.69	0.69
Placement	0.98	0.98	0.98	0.98
breast_cancer	0.95	0.95	0.95	0.95
Con._sonar_mines_vs_rocks_data	0.83	0.85	0.84	0.85
digits_dataset	0.99	0.99	0.99	0.99
german_credit	0.70	0.71	0.70	0.71
glass_identification	0.72	0.72	0.73	0.72
heart_disease	0.50	0.53	0.51	0.53
ionosphere	0.89	0.91	0.91	0.91
Iris	0.97	0.97	0.97	0.97
lung_cancer_data	0.47	0.53	0.48	0.53
lymphography	0.77	0.77	0.76	0.77
mushroom_data	0.98	0.98	0.98	0.98
spect_heart	0.80	0.79	0.80	0.79
tic_tac_toe_endgame	0.74	0.73	0.74	0.73
Titanic	0.75	0.76	0.76	0.77
wine_data	0.83	0.83	0.83	0.84
Zoo	0.98	0.99	0.99	0.99
Total Mean Accuracy	**0.80**	**0.82**	**0.81**	**0.82**

TABLE III
ACCURACY COMPARISON OF THE PROPOSED METHOD WITH OTHER CLASSIFIERS

Dataset	Proposed	KNN	DT	SVM	LR	NB	NN	AdaBoost	RF	GBost
ContactLens	0.69	0.60	0.78	0.60	0.71	0.81	0.76	0.78	0.76	0.70
Placement	0.98	0.98	0.98	0.69	0.83	1.00	0.60	0.98	0.95	0.98
breast_cancer	0.95	0.94	0.93	0.92	0.94	0.94	0.94	0.96	0.96	0.96
sonar_rocks	0.85	0.73	0.70	0.79	0.78	0.67	0.80	0.78	0.82	0.80
digits	0.99	0.98	0.85	0.99	0.96	0.84	0.97	0.29	0.97	0.95
german_credit	0.71	0.65	0.68	0.71	0.75	0.73	0.65	0.75	0.75	0.77
glass	0.72	0.66	0.70	0.33	0.61	0.43	0.44	0.45	0.78	0.74
heart	0.53	0.50	0.50	0.52	0.58	0.51	0.57	0.53	0.57	0.54
ionosphere	0.91	0.85	0.89	0.94	0.87	0.89	0.92	0.93	0.94	0.93
iris	0.97	0.97	0.95	0.96	0.97	0.95	0.96	0.93	0.96	0.95
lung_cancer	0.53	0.54	0.40	0.37	0.45	0.42	0.40	0.43	0.43	0.47
lymphography	0.77	0.76	0.76	0.80	0.81	0.67	0.82	0.65	0.81	0.86
mushroom	0.98	0.96	0.99	0.94	0.92	0.65	0.99	1.00	1.00	1.00
spect_heart	0.80	0.78	0.73	0.82	0.83	0.55	0.83	0.82	0.82	0.81
tic_tac_toe	0.74	0.82	0.86	0.86	0.67	0.71	0.82	0.76	0.93	0.92
titanic	0.77	0.71	0.77	0.67	0.80	0.79	0.79	0.80	0.81	0.81
wine	0.84	0.69	0.91	0.69	0.93	0.97	0.69	0.84	0.98	0.94
zoo	0.99	0.89	0.96	0.93	0.96	0.97	0.97	0.70	0.97	0.96
Mean Accuracy	**0.82**	**0.78**	**0.80**	**0.75**	**0.80**	**0.75**	**0.77**	**0.74**	**0.85**	**0.84**

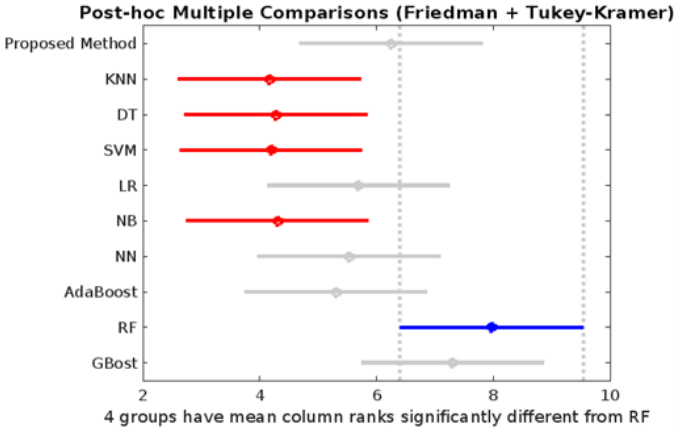

Fig. 4. Friedman Test Comparison between Classifiers.

IV. DISCUSSION

From Table III, it can be seen that the proposed algorithm outperforms both standard and advanced classification methods, including Decision Tree, KNN, and Logistic Regression, as well as more complex models such as Naive Bayes, MLP, SVM, and AdaBoost, in terms of accuracy.

Indeed, the average accuracy achieved across 18 datasets is **82%**, demonstrating the algorithm's robustness when faced with datasets that vary in size, complexity, and nature.

On the other hand, ensemble methods—namely Gradient Boosting and Random Forest—are the only techniques that outperform the proposed method. This can be explained by the fact that these methods are more complex and rely on multiple components (e.g., feature selection combined with several classifiers) to perform classification.

More specifically, our algorithm, which can be seen as a generalization of the traditional KNN method, outperforms KNN on **all tested datasets**.

Moreover, our method achieved the best results on **8 out of 18 datasets** and produced competitive performance on 8 other datasets, confirming its general effectiveness.

However, certain datasets such as *Contact Lens* and *Lung Cancer* highlight the limitations of our algorithm. In these cases, Naive Bayes or KNN achieved better results. This underperformance may be attributed to class imbalance or problem simplicity, which tend to favor probabilistic models or simpler algorithms.

The Friedman test (Figure 4) summarizes the results graphically, highlighting the superiority of the proposed method over classical classifiers, and also showing that only ensemble methods outperform it—primarily for the reason mentioned earlier.

V. CONCLUSION

This paper introduces a geometric extension of the K-Nearest Neighbors (KNN) classifier that leverages intra-class line segments and orthogonal projections to better model class structures and refine decision boundaries—especially in complex or noisy datasets.

A conflict removal step further enhances robustness by eliminating ambiguous segments.

Experimental results on 18 benchmark datasets show that the proposed method consistently surpasses the basic KNN and outperforms traditional classifiers such as SVM, Decision Tree, Logistic Regression, Naïve Bayes, and AdaBoost.

Future work will focus on incorporating adaptive voting and integrating the approach into ensemble frameworks for enhanced scalability and performance.

REFERENCES

[1] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. IT-13, no. 1, pp. 21–27, 1967.

[2] S. A. Dudani, "The Distance-Weighted K-Nearest-Neighbor Rule," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 325–327, 1976.

[3] R. D. Short and K. Fukunaga, "The Optimal Distance Measure for Nearest Neighbor Classification," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 622–627, 1981.

[4] C. Domeniconi, J. Peng, and D. Gunopulos, "Locally Adaptive Metric Nearest Neighbor Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1281–1285, 2002.

[5] T. Bailey and A. K. Jain, "A Note on Distance-Weighted K-Nearest Neighbor Rules," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 4, pp. 311–313, 1978.

[6] W. Zuo, K. Wang, H. Zhang, and D. Zhang, "Kernel Difference-Weighted k-Nearest Neighbors Classification," in *Advanced Intelligent Computing Theories and Applications. ICIC 2007*, LNCS, vol. 4682, Springer, Berlin, Heidelberg, pp. 759–766, 2007.

[7] J. M. Keller, M. R. Gray, and J. A. Givens, Jr., "A Fuzzy K-Nearest Neighbor Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 4, pp. 580–585, 1985.
[8] L. I. Kucnehva, "An Intuitionistic Fuzzy K-Nearest Neighbors Rule," *Notes on Intuitionistic Fuzzy Sets*, vol. 1, pp. 56–60, 1995.
[9] M.-S. Yang and C.-H. Chen, "On the Edited Fuzzy K-Nearest Neighbor Rule," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 28, no. 3, pp. 461–466, 1998.
[10] J. Hu, H. Peng, J. Wang, and W. Yu, "kNN-P: A kNN Classifier Optimized by P Systems," *Theoretical Computer Science*, vol. 817, pp. 55–65, 2020.
[11] J. Zhang, Y. Niu, and W. He, "Using Genetic Algorithm to Improve Fuzzy KNN," in *Proc. Int. Conf. Computational Intelligence and Security (CIS)*, vol. 1, pp. 475–479, 2008.
[12] H. Yigit, "A Weighting Approach for KNN Classifier," in *Proc. Int. Conf. Electronics, Computer and Computation (ICECCO)*, pp. 228–231, 2013.
[13] C.-Y. Lee, K.-Y. Huang, Y.-X. Shen, and Y.-C. Lee, "Improved Weighted k-Nearest Neighbor Based on PSO for Wind Power System State Recognition," *Energies*, vol. 13, no. 20, pp. 5520, 2020.