KNN CLAS

Eduardo Henrique Basilio de Carvalho
Departamento de Engenharia Eletrônica
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
eduardohbc@ufmg.br

***Abstract*—This document proposes and analyzes an alternative to the NN-CLAS method proposed by Arias-Garcia et al. (2021). The KNN-CLAS leverages the voting mechanism of the K-nearest neighbors (KNN) classifier to eliminate the computationally expensive filtering step required during the training phase of NN-CLAS. The original NN-CLAS relies on Gabriel graph computations and a vertex-degree-based noise filtering process to identify structural support edges (SEs) for classification. In contrast, KNN-CLAS bypasses this filtering by aggregating decisions from multiple neighbors, thereby reducing training complexity while maintaining competitive accuracy. Experimental results on benchmark datasets demonstrate that the proposed method achieves comparable performance to NN-CLAS, with significant efficiency improvements. The methodology is particularly suited for embedded systems due to its reduced computational overhead and parameter-free design.**

***Index Terms*—pattern recognition, large margin classifiers, Gabriel graph, KNN classifier, embedded systems**

I. Introduction

Large margin classifiers, such as support vector machines (SVMs), rely on optimization techniques to maximize separation between classes. However, their computational complexity and dependency on user-defined parameters limit their applicability in embedded systems. The NN-CLAS framework, introduced by Torres et al. (2016), addresses these limitations by constructing classifiers directly from the geometric structure of the training data using Gabriel graphs (GGs). The GG encodes pairwise relationships between data points, and support edges (SEs) connecting vertices of opposing classes define local hyperplanes that collectively form a large-margin decision boundary [1]. While effective, NN-CLAS requires a filtering step to remove noisy vertices, which involves evaluating the quality of each vertex based on its neighborhood structure. This process, though critical for robustness, incurs significant computational costs, especially for large datasets [2], [3].

The proposed KNN-CLAS eliminates the need for explicit filtering by leveraging the inherent noise resilience of the KNN voting mechanism. Instead of pruning the dataset during training, KNN-CLAS directly uses the GG's SEs and assigns class labels through a majority vote among the nearest neighbors. This approach retains the structural benefits of GG-based classification while simplifying the training pipeline. The remainder of this section details the original NN-CLAS filtering methodology and its computational challenges.

A. Filtering

The NN-CLAS framework constructs a Gabriel graph G_G from the training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$, where edges connect vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ if no other point lies within the hypersphere defined by their diameter [1]. To handle overlapping classes, a quality measure $q(\mathbf{x}_i)$ evaluates the ratio of same-class neighbors to total neighbors for each vertex:

$$q(\mathbf{x}_i) = \frac{\hat{\mathcal{A}}(\mathbf{x}_i)}{\mathcal{A}(\mathbf{x}_i)},$$

where $\mathcal{A}(\mathbf{x}_i)$ is the vertex degree and $\hat{\mathcal{A}}(\mathbf{x}_i)$ counts neighbors sharing $\mathbf{x}_i$'s class label [2]. Vertices with $q(\mathbf{x}_i)$ below class-specific thresholds t^+ and t^-—calculated as the mean quality per class—are discarded as noise. This filtering ensures SEs lie near the true class boundaries but requires $O(n^2)$ distance computations and iterative quality evaluations, making it impractical for resource-constrained systems [3]. The KNN-CLAS circumvents this bottleneck by integrating neighbor voting, thereby avoiding explicit structural filtering while preserving classification accuracy.

B. KNN-CLAS

The KNN-CLAS method simplifies the NN-CLAS framework by eliminating the filtering step and leveraging the K-nearest neighbors (KNN) voting mechanism for classification. The methodology can be summarized as follows:

- **K-Nearest Neighbors**: For a given test sample $\mathbf{z}_j$, compute the distances to all training samples $\mathbf{x}_i$ and identify the k-nearest neighbors, denoted as $\mathcal{N}_k(\mathbf{z}_j)$.
- **Weighted Voting**: Assign weights to the neighbors using a kernel function $K(d)$, such as the Gaussian kernel $K(d) = e^{-d}$, where d is the distance between $\mathbf{z}_j$ and $\mathbf{x}_i$.
- **Decision Sum**: Compute the decision sum for $\mathbf{z}_j$ by aggregating the weighted contributions of the neighbors' class labels:

$$S(\mathbf{z}_j) = \sum_{(\mathbf{x}_i, y_i) \in \mathcal{N}_k(\mathbf{z}_j)} y_i \cdot K(d(\mathbf{z}_j, \mathbf{x}_i)).$$

- **Classification**: Assign the class label $\hat{y}_j$ based on the sign of the decision sum:

$$\hat{y}_j = \text{sign}(S(\mathbf{z}_j)).$$

This approach bypasses the computationally expensive filtering step of NN-CLAS while maintaining competitive accuracy. By directly aggregating decisions from multiple neighbors, KNN-CLAS reduces training complexity and is particularly suited for embedded systems with limited computational resources.

II. Methodology

A. Dataset Metadata

TABLE I
DATASET METADATA

Dataset	Samples	Features
Ionosphere	351	34
Binary Digits	360	64
Haberman	306	3
Pima Diabetes	768	8
Banknote	1372	4
Sonar	208	60
Breast Cancer	569	30
SPECT Heart	349	44

Table I provides an overview of the datasets used in this study, including the number of samples and features for each dataset. These characteristics are critical for understanding the computational complexity and scalability of the proposed KNN-CLAS method. For instance, datasets with a higher number of features, such as Binary Digits and Sonar, may require more computational resources during distance calculations, while smaller datasets like Haberman and SPECT Heart are less demanding.

The diversity in dataset sizes and feature dimensions ensures a comprehensive evaluation of the proposed method across various scenarios. This metadata also highlights the challenges posed by datasets with fewer features, such as Pima Diabetes and Banknote, where class separability may be more dependent on the quality of the decision boundary.

III. Results

A. Accuracy Comparison

TABLE II
MODEL ACCURACY COMPARISON

Dataset	Accuracy			
	nn	1nn	3nn	5nn
Ionosphere	0.88	0.86	0.89	0.88
Binary Digits	1.00	1.00	1.00	1.00
Haberman	0.69	0.69	0.71	0.70
Pima Diabetes	0.75	0.68	0.71	0.74
Banknote	1.00	1.00	1.00	1.00
Sonar	0.73	0.89	0.89	0.87
Breast Cancer	0.95	0.95	0.96	0.96
SPECT Heart	0.68	0.86	0.85	0.82

Table II shows the accuracy of NN-CLAS and KNN-CLAS across various datasets. KNN-CLAS achieves comparable accuracy to NN-CLAS, with slight improvements in datasets such as Sonar and SPECT Heart.

B. Training and Prediction Times

TABLE III
TRAINING AND PREDICTION TIMES

Dataset	Training (ms)		Prediction (ms)			
	nn	knn	nn	1nn	3nn	5nn
Ionosphere	89.80	26.40	2.70	2.90	2.90	2.70
Binary Digits	219.70	68.80	2.90	3.00	3.00	2.90
Haberman	16.80	6.30	2.30	3.00	3.10	3.60
Pima Diabetes	80.10	28.80	2.40	3.20	3.00	3.10
Banknote	434.70	75.00	4.00	4.60	4.60	4.40
Sonar	128.50	45.80	6.40	6.60	5.70	6.30
Breast Cancer	211.20	31.20	2.60	3.00	2.90	3.00
SPECT Heart	431.60	136.90	5.20	5.40	5.30	5.60

Table III highlights the significant reduction in training time achieved by KNN-CLAS compared to NN-CLAS. For instance, in the Banknote dataset, KNN-CLAS reduces training time from 434.70 ms to 75.00 ms, making it more suitable for embedded systems.

C. Support Samples

TABLE IV
SUPPORT SAMPLES COUNT

Dataset	Support Samples	
	nn	knn
Ionosphere	111	252
Binary Digits	124	268
Haberman	37	274
Pima Diabetes	127	594
Banknote	160	197
Sonar	94	186
Breast Cancer	101	344
SPECT Heart	100	282

Table IV presents the number of support samples used by each method. KNN-CLAS generally requires more support samples, which may increase memory usage but simplifies the training process.

D. Dataset Characteristics

TABLE V
DATASET STATISTICS

Dataset	C0/C1	MI	Fisher	Overlap	Imb.Ratio
Ionosphere	0.56	0.21	0.11	0.85	1.79
Binary Digits	0.98	0.18	1.39	0.44	1.02
Haberman	0.36	0.03	0.07	1.0	2.78
Pima Diabetes	0.54	0.05	0.18	0.88	1.87
Banknote	0.8	0.19	0.7	0.75	1.25
Sonar	1.14	0.03	0.09	1.0	1.14
Breast Cancer	1.68	0.21	1.03	0.47	1.68
SPECT Heart	2.67	0.07	0.18	0.91	2.67

Table V summarizes the dataset characteristics, including class imbalance and feature overlap. These factors influence the performance of both methods, with KNN-CLAS showing robustness to class imbalance.

E. Likelihood

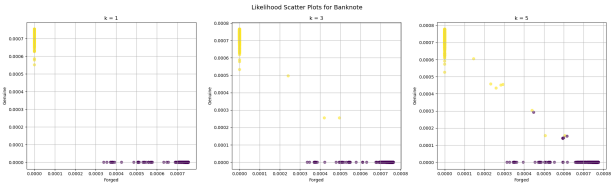

Fig. 1. Likelihood comparison for the Banknote dataset.

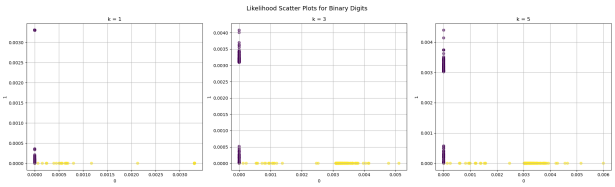

Fig. 2. Likelihood comparison for the Binary Digits dataset.

Fig. 3. Likelihood comparison for the Breast Cancer dataset.

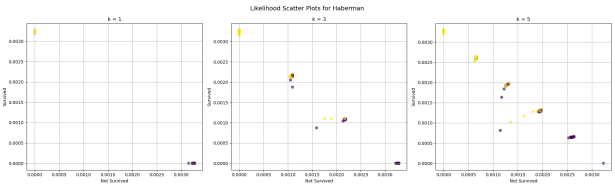

Fig. 4. Likelihood comparison for the Haberman dataset.

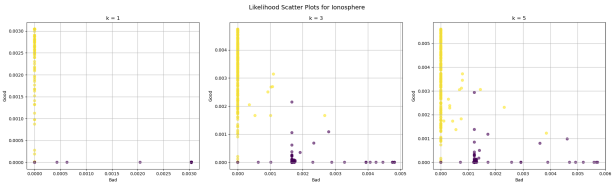

Fig. 5. Likelihood comparison for the Ionosphere dataset.

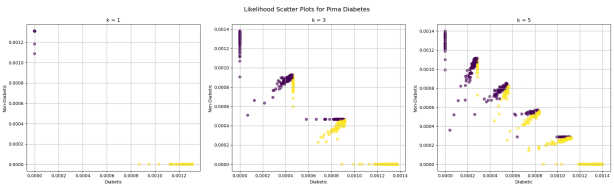

Fig. 6. Likelihood comparison for the Pima Diabetes dataset.

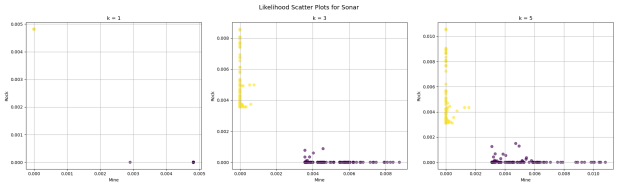

Fig. 7. Likelihood comparison for the Sonar dataset.

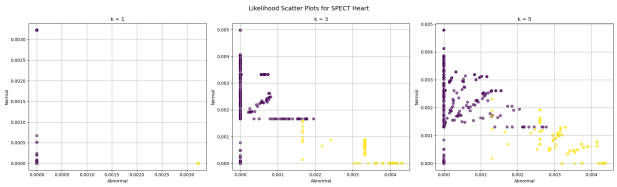

Fig. 8. Likelihood comparison for the SPECT Heart dataset.

IV. Discussion

The results demonstrate that KNN-CLAS offers a practical alternative to NN-CLAS, particularly for resource-constrained environments. While NN-CLAS relies on computationally expensive filtering, KNN-CLAS leverages the KNN voting mechanism to achieve similar accuracy with reduced training times. However, the increased number of support samples in KNN-CLAS may pose challenges for memory-constrained systems.

The performance of KNN-CLAS is influenced by dataset characteristics such as class imbalance and feature overlap. For example, in datasets with high imbalance ratios (e.g., Haberman), KNN-CLAS maintains competitive accuracy, showcasing its robustness.

Future work could explore hybrid approaches that combine the strengths of both methods, such as selective filtering to reduce support samples while retaining efficiency.

Acknowledgment

References

[1] L. C. B. Torres, "Classificador por arestas de suporte (CLAS): métodos de aprendizado baseados em Grafos de Gabriel," Manuscript, 2016.

[2] A. C. Souza, C. Leite Castro, J. A. Garcia, L. C. B. Torres, L. J. Acevedo Jaimes and B. R. A. Jaimes, "Improving the Efficiency of Gabriel Graph-based Classifiers for Hardware-optimized Implementations," 2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA), Bucaramanga, Colombia, 2019.

[3] J. Arias-Garcia et al., "Enhancing Performance of Gabriel Graph-Based Classifiers by a Hardware Co-Processor for Embedded System Applications," in IEEE Transactions on Industrial Informatics, vol. 17, no. 2, Feb. 2021.

[4] J. Arias-Garcia et al., "Improved Design for Hardware Implementation of Graph-Based Large Margin Classifiers for Embedded Edge Computing," in IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 1, Jan. 2024.

[5] L. C. B. Torres, C. L. Castro and A. P. Braga, "A parameterless mixture model for large margin classification," 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015.

[6] L. C. B. Torres, C. L. Castro, F. Coelho and A. P. Braga, "Large Margin Gaussian Mixture Classifier With a Gabriel Graph Geometric Representation of Data Set Structure," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 3, March 2021.

[7] L. C. B. Torres, C. L. Castro, F. Coelho, F. Sill Torres and A. P. Braga, "Distance-based large margin classifier suitable for integrated circuit implementation," Manuscript, 2015.
[8] D. Dua and C. Graff, "Breast Cancer Wisconsin (Diagnostic) Data Set," UCI Machine Learning Repository, 1995. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
[9] J. Brownlee, "Pima Indians Diabetes Dataset," GitHub Repository, 2020. [Online]. Available: https://github.com/jbrownlee/Datasets
[10] D. Dua and C. Graff, "Haberman's Survival Data Set," UCI Machine Learning Repository, 1995. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Haberman's+Survival
[11] D. Dua and C. Graff, "Data Banknote Authentication Data Set," UCI Machine Learning Repository, 1995. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/banknote+authentication
[12] D. Dua and C. Graff, "Connectionist Bench (Sonar, Mines vs. Rocks) Data Set," UCI Machine Learning Repository, 1995. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/connectionist+bench+(sonar,+mines+vs.+rocks)
[13] D. Dua and C. Graff, "Adult Data Set," UCI Machine Learning Repository, 1995. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/adult
[14] D. Dua and C. Graff, "Ionosphere Data Set," UCI Machine Learning Repository, 1995. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/ionosphere
[15] D. Dua and C. Graff, "SPECT Heart Data Set," UCI Machine Learning Repository, 1995. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SPECT+Heart
[16] L. Breiman et al., "Optical Recognition of Handwritten Digits Data Set," Scikit-learn Documentation, 1998. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html