

Article

Bipolar Metric Spaces for Geometric Regularization in Distance-Based Learning

Gizen Mutlu *  and Çiğdem İnan Acı 

Department of Computer Engineering, Faculty of Engineering, Mersin University, Mersin 33150, Türkiye; caci@mersin.edu.tr

* Correspondence: gmutlu@mersin.edu.tr

Abstract

Distance-based machine learning algorithms are widely used due to their simplicity and flexibility. However, they often suffer from sensitivity to noise, high computational cost, and instability in high-dimensional and complex data spaces. To address these limitations, this study introduces a novel geometric learning framework based on Bipolar Metric Spaces (BMSs). The proposed framework extends classical metric structures by incorporating a quadrilateral inequality constraint, enabling the modeling of both similarity and contrast relationships across data points. Within this framework, geometric regularization is achieved through a structural filter mechanism that filters out structurally inconsistent assignments and stabilizes decision boundaries. Importantly, the proposed approach is not limited to specific algorithms and can be integrated into a broad class of distance-based learning methods. To demonstrate its effectiveness, the framework is instantiated on K-Nearest Neighbor (KNN) and K-means algorithms, resulting in Bipolar K-Nearest Neighbor (BP-KNN) and Bipolar K-means (BP-K-means). Theoretical analysis shows that the proposed framework improves Lipschitz stability, reduces variance, and enhances generalization performance. The models were tested on six different datasets, including benchmark datasets (Iris, Wine, and Breast Cancer) and real-world datasets (Abalone, CDC Diabetes, and Malicious Website Features (MWS)). Theoretical complexity analysis, supported by Floating Point Operation (FLOP) quantification, reveals that the bipolar approach offers a more stable decision boundary and significantly faster convergence, corroborating the observed efficiency improvements. Notably, on the large-scale MWS dataset, BP-KNN achieved an 11.6-fold increase in inference speed and a 75% reduction in RAM usage compared with the standard KNN. These results demonstrate that BMS provides a unified and scalable geometric regularization framework for distance-based learning, offering a principled solution to the trade-off between accuracy, robustness, and computational efficiency.

Keywords: Bipolar Metric Space (BMS); distance metric; structural consistency; KNN; K-means; computational efficiency; machine learning

MSC: 54E35 (primary); 68T05, 68T10, 62H30 (secondary).



Academic Editor: Valsamis Ntouskos

Received: 28 April 2026

Revised: 28 June 2026

Accepted: 14 July 2026

Published: 17 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Distance-based machine learning algorithms, notably K-Nearest Neighbors (KNN) and K-means, are foundational due to their simplicity, flexibility, and extensive applicability across diverse data types. These algorithms are employed to identify similarities by

assessing the proximity of data points [1]. Despite their prevalent use, these methods encounter challenges such as high memory consumption, inefficiency in processing time, and considerable sensitivity to noise and interclass overlap, particularly in large datasets [2–4]. The primary limitations of the KNN and K-means algorithms stem from their reliance on metrics that consider only pairwise relationships within a single dataset. This reliance inhibits their ability to fully capture the intricate geometric integrity of data structures [5]. Additionally, their performance often deteriorates when applied to high-dimensional, unstable, or geometrically complex datasets [6]. This issue underscores the need for a novel, robust framework capable of more effectively and efficiently capturing similarity relationships in data, ensuring structural consistency, and enhancing the generalization capabilities of these traditional algorithms. Numerous methodologies have been proposed in the literature to address these issues, encompassing various approaches such as prototype selection, local feature weighting, fast search strategies, metric learning, and data complexity measures [7–12]. Prototype selection and local feature weighting are designed to enhance classification speed and accuracy while minimizing unnecessary data storage and class overlap. In contrast, fast search methods based on optimal triangle inequality aim to reduce distance computation costs while preserving full-neighborhood relationships [7,11]. Studies based on metric learning, however, enhance flexibility and performance through new loss functions that mitigate the impact of outliers, dynamic neighborhood selection, class-based confidence measures, and similarity measures that are more robust to noise [13,14]. Consistent with the KNN literature, K-means and its derivatives, another significant area of distance-based methods, have similarly concentrated on performance improvement. The standard K-means algorithm has limitations, such as the need to predetermine the number of clusters, random selection of initial centers, low accuracy in complex data structures, and difficulty adapting to varying cluster densities and shapes [15]. Methods developed to address these limitations include fixed-center clustering approaches, hybrid clustering strategies, new distance metrics utilizing derivative information, deterministic optimization techniques for initial center selection, and subspace learning methods that also consider the separation between clusters [16–21]. An analysis of the existing literature reveals a significant research gap in the optimization of distance-based machine learning algorithms. While recent studies have made considerable progress in improving the KNN and K-means algorithms through prototype selection [11], rapid search strategies using the triangle inequality [7], and advanced metric learning frameworks [10,12], a common limitation remains: these approaches primarily concentrate on unipolar distance, that is, the direct distance between two points. These optimizations evaluate similarity solely through direct pairwise relationships within a single data space, thus overlooking the broader geometric context, indirect pathways, and opposing interactions among data points. Ultimately, the contemporary literature on optimizations based on the triangular inequality focuses exclusively on the speed. In contrast, our proposed framework serves as both a speed and structural noise filter, thereby stabilizing the decision boundaries.

Conversely, advanced metric learning models [8,12] improve accuracy and robustness against noise but typically introduce substantial computational overhead, rendering them impractical for large-scale or high-dimensional datasets. A unified mathematical framework is absent, which inherently necessitates spatial consistency to concurrently optimize classification accuracy, algorithm complexity, and memory usage. To address this gap, this study proposes an innovative framework based on the Bipolar Metric Space (BMS) [22], which derives the quadrilateral inequality from the traditional triangle inequality. Unlike existing methods that seek to ameliorate the deficiencies of classical algorithms through local feature weighting or dynamic neighbor selection, the proposed framework fundamentally redesigns the distance measurement. By introducing a geometric constraint that

models both polar similarities and contrasts, the proposed approach rigorously filters out geometrically inconsistent assignments. This geometric configuration aims to transcend the traditional accuracy–efficiency trade-off, offering a theoretically robust and practically efficient solution in terms of memory consumption, execution time, and stability.

In conclusion, this study introduces a geometric filtering framework positioning BMS as a structural operator applicable to distance-based learning. We introduce an innovative BMS-based framework for the traditional KNN and K-means algorithms. This methodology calculates distance measurements based on absolute differences and pairwise contrast relationships, thereby fundamentally stabilizing decision boundaries. By offering a distinctive generalization of classical learning, the bipolar approach directly models the inherent polar similarity–contrast relationships within the data. The Bipolar K-Nearest Neighbors (BP-KNN) and Bipolar K-means (BP-K-means) algorithms overcome the limitations of traditional KNN and K-means algorithms by being responsive to sample coordinates and the direction of interaction between pairs. This bipolar sensitivity facilitates more balanced, stable, and highly generalizable outcomes for both classification and clustering tasks.

The remainder of this paper is organized as follows. Section 2 offers a comprehensive review of related studies on distance metric learning and algorithmic optimization. Section 3 establishes the theoretical groundwork for the BMS, not only including a mathematical proof of the BMS constraints but also detailing the BP-KNN and BP-K-means architectures and their respective algorithms. Section 4 presents the proposed framework’s experimental setup and provides an in-depth analysis of the performance results for the six datasets. Section 5 provides a conceptual discussion on computational complexity, Floating Point Operation (FLOP) reduction, and model limitations. Finally, Section 6 concludes the study and outlines the potential directions for future research.

2. Related Work

This section presents a comprehensive review of related studies that focus on improving the KNN and K-means algorithms with metric learning, prototype selection, and different clustering strategies.

2.1. KNN Approaches

To solve the fundamental problems of the KNN algorithm, such as interclass overlap, high memory requirements, and parameter selection, Zhang et al. [11] developed the Improved Prototype Selection and Local Feature-Weighted KNN (IKNN_PSLFW) method. This approach divides the training data into pure subsets and selects the most distinctive example as the prototype for each cluster, stores the information of the unselected examples along with the range and number of examples, and reduces the effect of noise by assigning a high weight to low-variance features. Owing to the dynamic classification mechanism that eliminates the need to predetermine the k parameter, the model exhibited superior performance compared to the traditional KNN in terms of both accuracy and time efficiency. Pan et al. [7] proposed Optimal Triangle Inequality (OTI), which provides a solution to the high-computational-cost problem of KNN by preserving full-neighborhood relationships, and Effective Optimal Triangle Inequality (EOTI) strategies with lower memory cost; thus, the elimination performance especially in marginal regions increased and the search time was significantly shortened. In the field of metric learning, Yuan et al. [12] developed two Mixture-Correntropy (MC)-based models (MCML and BMCML) with limited loss function features to address the problem of sensitivity to outliers and noise. These models were more stable and achieved higher accuracy rates than their competitors on both the clean and noisy datasets. Yang et al. [8] proposed the NMD model by integrating Neighborhood Rough Cluster (NRS) theory with Distance Metric Learning (DML), thus minimizing the

uncertainty region by defining the neighborhood with a data-specific learned metric. Ruan et al. [10] developed a model called DM-C2IC, which is not KNN-based and has a convex structure, and optimized both the Mahalanobis distance metric and class-based confidence measures simultaneously, guaranteeing convergence to the global optimum. In a previous study, Ruan et al. [9] introduced the NNS-DML model, which automatically determines the optimal number of neighbors for each sample, thus eliminating the dependence on the k parameter. Dai et al. [23] developed the Distance Map Overlap Index (DMOI) metric to measure class overlap quickly and efficiently, significantly accelerating complexity measurement in large datasets.

2.2. K-Means Approaches

Studies on K-means and its derivatives have focused on addressing the fundamental limitations of the algorithm. Ay et al. [21] proposed the FC-Kmeans and FC-Kmeans 2 algorithms, which provide fixed-center clustering for scenarios in which some centers are immutable, and showed that these methods can perform with statistically similar accuracy to standard K-means++. Laskhmaiah et al. [24] developed the NOC-OKMDDC algorithm, which includes CNN-based feature reduction, optimized K-means segmentation, and density-based merging stages to satisfy the non-overlapping constraint in high-dimensional spatial data. This method achieved superiority over its competitors in terms of the Adjusted Rand Index values. Abdulnassar and Nair [25] developed the Kmeans9+ model using deterministic starting center algorithms to reduce the randomness and computational costs in estimating the k value and selecting the starting center. They increased efficiency by reducing the number of comparisons by $9/k$. To enhance the efficiency of exact K-means, Xia et al. [26] introduced the Ball k-means algorithm, which employs ball structures to characterize clusters and divides them into stable and active regions. By precisely identifying neighboring clusters, this method limits distance computations to a select few candidate centroids, thereby achieving sublinear time complexity in subsequent iterations without the necessity of maintaining upper or lower bounds. Although this approach is highly efficient for large-scale datasets, its performance is contingent upon the unipolar proximity between data points and cluster centers. Consequently, it may not fully address structural inconsistencies within complex data manifolds.

A comprehensive review of the existing literature (Table 1) has identified several notable research gaps. Advanced metric learning models, such as MCML, BMCML, and NNS-DML [9,10,12], have been shown to enhance classification accuracy and noise resistance. However, complexity analyses reveal that these models introduce significant computational overhead, such as cubic complexity ($\mathcal{O}(N^3)$) or dense matrix operations, rendering them impractical for large-scale datasets like the Malicious Website Features (MWS) data used in this study. While rapid search strategies, such as OTI/EOTI [7] and the sublinear Ball k-means algorithm [26], effectively reduce distance computation costs through aggressive pruning, their optimization is primarily speed-oriented. Although Ball k-means achieves notable acceleration by partitioning clusters into stable and active areas, it remains vulnerable to the curse of dimensionality and does not explicitly filter local structural noise during the pruning process. Therefore, a robust pruning strategy should serve as both a speed-enhancing mechanism and a structural noise filter to stabilize decision boundaries. Current K-means optimizations often depend on rigid spatial partitioning and complex hybrid pipelines. For example, although the NOC-OKMDDC model [24] integrates deep learning and swarm optimization to manage diverse densities, its multistage architecture introduces considerable algorithmic complexity and sensitivity to manual penalty tuning. Other constrained or initialization-based approaches [21,25] face challenges with boundary oscillations due to the lack of a stringent geometric fil-

ter to maintain cluster structural integrity. Despite these advances, none of the existing methods enforce a global geometric consistency constraint that simultaneously reduces noise sensitivity and computational redundancy. Also, in these methods, similarity was assessed solely through direct unipolar binary relationships within a single data space. This narrow focus overlooks the broader geometric context, indirect relational paths, and opposing interactions among data points. To address these limitations, this study shifts the optimization focus from the triangle inequality to the quadrilateral inequality, with the BP-KNN and BP-K-means algorithms addressing these gaps. Unlike previous studies, our approach employs a quadrilateral inequality-inspired structural constraint. This not only provides structural noise robustness but also reduces computational redundancy through an efficient candidate-pruning mechanism. Thus, it achieves a critical balance in high-dimensional and resource-constrained environments by efficiently filtering out structurally flawed points, a feature not found in previous studies.

Table 1. The summary of the studies.

Author and Year	Category	Method	Best Result	Limitations
Pan et al., 2020 [7]	Fast Nearest Neighbor Search	OTI & EOTI	RRDC 73.23%	High space complexity and extra computational overhead for triangle check strategies.
Xia et al., 2020 [26]	Accelerated Exact K-means	Ball k-means	144.3x speedup (Birch dataset)	Efficiency degrades in very high-dimensional spaces.
Yang et al., 2021 [8]	Metric-integrated Rough Sets	EFNMD/ FNMD	83.06% (EFNMD avg.)	High computational overhead and memory consumption due to the DML process, limiting scalability for big data.
Laskhmaiah et al., 2021 [24]	Hybrid K-means Optimization	NOC-OKMDDC	81 Adjusted Rand Index	High architectural complexity due to its multi-stage hybrid pipeline and sensitivity to manual penalty tuning.
Ruan et al., 2021 [9]	Metric Learning	NNS-DML	0.975 (Vote)	Suffers from high computational complexity ($\mathcal{O}(N^2)$ per iteration), which limits its scalability and structural robustness in large datasets.
Ay et al., 2023 [21]	Constrained K-means Clustering	FC-Kmeans/ FC-Kmeans2	0.76 Silhouette index	Performance significantly degrades when the fixed center ratio exceeds 50%, with limited adaptability to boundary noise.
Abdulnassar and Nair, 2023 [25]	K-means Initialization and Search Optimization	Kmeans9+	Iris: 99% Accuracy	Designed for uniform data, struggles with non-uniform distributions and complex cluster geometries.
Yuan et al., 2024 [12]	Kernel Metric Learning	MCML/ BMCML	98.31% (COIL)	High computational cost per iteration ($\mathcal{O}(N^3)$) and sensitivity to kernel width tuning.
Dai et al., 2024 [23]	Overlapping Complexity Measurement	DMOI	Highest correlation with -0.9316	Sensitive to mapping bias from pseudo-centers in noisy data and highly dependent on the chosen distance metric.

Table 2 further clarifies the methodological positioning of BP-KNN compared with the advanced distance-based learning approaches discussed in related studies. While OTI/EOTI primarily aims at a fast nearest neighbor search via triangle inequality-based elimination, MCML/BMCML and NNS-DML focus on learned similarity structures and dynamic neighborhood generation. In contrast, BP-KNN is designed as an add-on geometric arrangement operator that implements a BMS-inspired bipolar sign-space structural filter before the final numerical KNN sorting. Therefore, instead of learning a metric intensively or optimizing the neighborhood structure itself, the proposed method reduces the final full distance sort workload by generating the retained candidate set Ω_i . This comparison aims to clarify the mechanism, computational balance, and scalability of BP-KNN.

Table 2. Extended positioning of BP-KNN relative to advanced distance-based learning methods.

Method Category	Representative Methods	Main Objective	Main Computational Bottleneck	Relation to BP-KNN
Fast search strategies	OTI/EOTI	Reduce nearest-neighbor distance computations using triangle-inequality-based elimination.	Additional bound checking, memory overhead, and dependency on triangle-based pruning structure.	Primarily speed-oriented; BP-KNN instead applies a bipolar sign-space structural filter before final numerical ranking.
Metric learning	MCML/BMCML	Learn a more robust similarity or metric structure under noise and outliers.	Dense metric/kernel operations, optimization cost, and high scalability burden on large datasets.	Accuracy/robustness-oriented; BP-KNN does not learn a dense metric but applies a lightweight structural filter.
Dynamic neighborhoods	NNS-DML	Adaptively determine the neighborhood structure and reduce dependence on fixed k .	Repeated neighborhood and metric optimization, typically expensive for large-scale data.	Addresses k -sensitivity; BP-KNN instead uses validation-selected k and a structural candidate filter.
Proposed framework	BP-KNN	Apply a BMS-inspired bipolar structural filter before final KNN ranking.	Candidate scanning plus sign-space structural validation.	Designed as a plug-in structural regularizer for distance-based learning; reduces final full-distance evaluations from n to $ \Omega_i $.

3. Bipolar Learning Framework

In this study, we introduce a unified geometric filtering framework by positioning BMS as a structural operator for distance-based learning. The overall architecture of the proposed framework, which serves as a common foundation for both BP-KNN and BP-K-means, is illustrated in Figure 1. We generalize the proposed approach as a bipolar extension of distance-based learning algorithms:

$$\mathcal{A}_{\text{bipolar}} = \mathcal{A}_{\text{standard}} + \mathcal{C}_{(B3)}. \quad (1)$$

This formulation implies that the proposed framework is not limited to specific algorithms such as KNN or K-means, but can be applied to a broad class of distance-based methods. In BMS, the (B3) constraint acts as a structural regularizer, improving robustness and generalization. The BP-KNN and BP-K-means algorithms transition the focus of optimization from a unipolar triangle inequality to a bipolar quadrilateral inequality. Utilizing the fixed-point theorems established by [22], we demonstrate an acceleration in convergence and an improvement in error bounds.

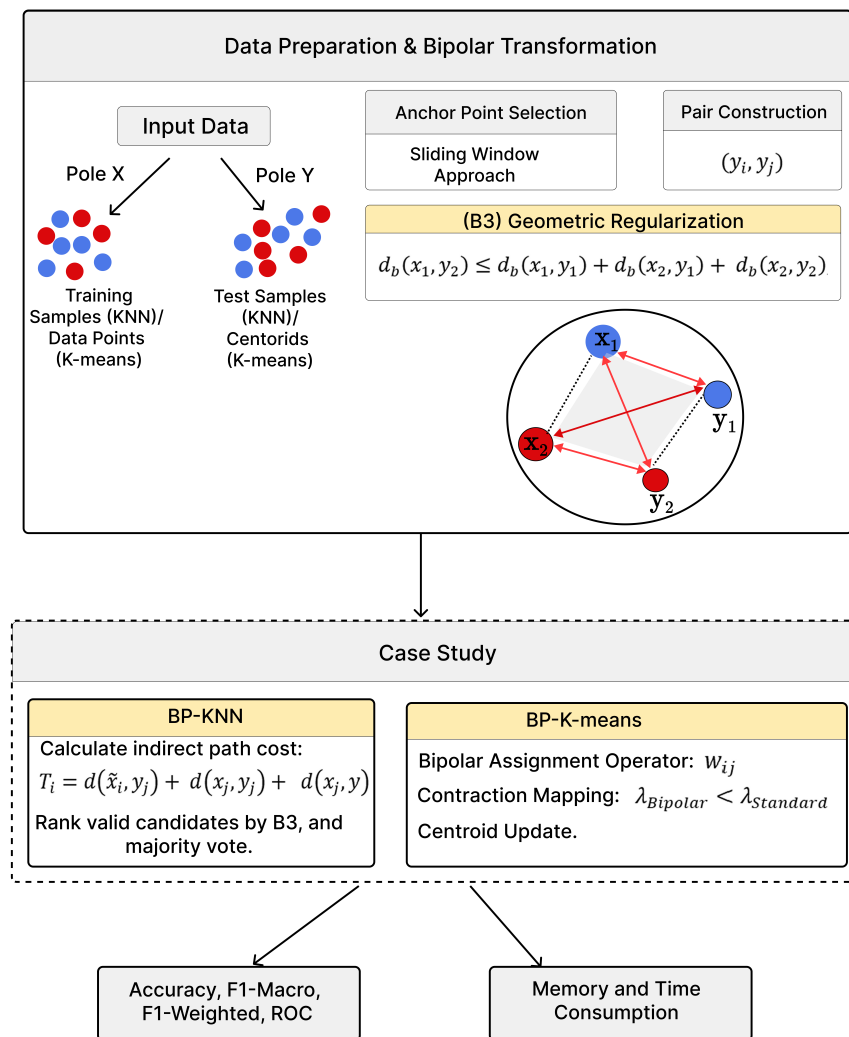


Figure 1. Architecture of the proposed bipolar framework.

3.1. Bipolar Metric Spaces (BMS)

Classic metric spaces focus on the distances between points within a single set, where distances between points of the same type may be either undefined or incalculable due to insufficient data. Conversely, when distances are considered between elements of two distinct sets rather than within a single set, they are addressed as isometrics and are generalized under the BMS [22]. In scenarios involving two distinct sets of objects X , and Y rather than reducing a measure such as congruence, similarity, or dissimilarity to a single set, the BMS can be directly defined on $X \times Y$. Unlike the classical metric applicable to a single set, the bipolar metric concept operates with distances defined on $X \times Y$. The primary objective is to formulate BMS axioms and pseudo-metrics from these spaces to more accurately represent similarities and dissimilarities by establishing convergence properties. Within the BMS framework, elements from two different sets are directly compared, thereby generating lateral information about the internal structure of the sets. Specifically, if two elements from X exhibit similar behavior with respect to the entirety of Y , they are considered similar. In the same way, if two elements from Y exhibit similarly behavior with respect to X , an intuitive understanding is achieved [22]. The conditions that satisfy the definition of a bipolar metric are as follows. Let X and Y be non-empty sets

and let $d_b : X \times Y \rightarrow \mathbb{R}^+$ be a function representing the set of non-negative real numbers. Accordingly, the following properties are considered [22].

- (B0) If $d_b(x, y) = 0$, then $x = y$ for all $(x, y) \in X \times Y$.
- (B1) If $x = y$ then $d_b(x, y) = 0$ for all $(x, y) \in X \times Y$.
- (B2) $d_b(x, y) = d_b(y, x)$ for all $(x, y) \in X \cap Y$.
- (B3) Quadrilateral inequality:

$$d_b(x_1, y_2) \leq d_b(x_1, y_1) + d_b(x_2, y_1) + d_b(x_2, y_2),$$
for all $x_1, x_2 \in X$ and $y_1, y_2 \in Y$.

The concept of a BMS is a generalized version of the classical metric space concept. More specifically, for bipolarity (X, Y, d_b) , if $X = Y$, then (X, d_b) and its equivalent (Y, d_b) are metric spaces. Thus, the quadrilateral inequality, which is condition (B3) in the definition of a BMS, is equivalent to the triangle inequality in metric spaces as $x_2 = y_2 \in X = Y$ [22]. The proposed approach should not be interpreted as a modification of specific algorithms such as KNN or K-means. Instead, it introduces a geometric filtering operator defined by the (B3) constraint. The BP-KNN algorithm incorporates BMS axioms, specifically condition (B3), into the decision process to overcome the limitations of the KNN, such as the influence of noisy neighbors. The goal was to ensure structural consistency and make more reliable decisions by considering geometric consistency rather than relying solely on direct distances. This approach mitigates the influence of misleading neighbors, particularly in regions where class boundaries are unclear, and provides a fairer voting mechanism for unbalanced datasets.

In the context of a BMS, a bipolar metric is concerned solely with the distances between points situated at two distinct poles of the space, thereby offering insights into the internal structure of these poles [22].

Definition 1 (Bipolar Geometry). *Let (X, Y, d) be a BMS. The pair (X, Y) together with the cross-domain distance $d : X \times Y \rightarrow \mathbb{R}^+$ induces a bipolar geometric structure, where the similarity between elements within the same domain is determined indirectly via their relational profiles over the opposite domain.*

Proposition 1 (Relational Consistency). *Let (X, Y, d) be a BMS and let d_X be defined as in Definition 1. Then, for any $x_1, x_2 \in X$, the quantity $d_X(x_1, x_2)$ measures the maximal discrepancy between their relational profiles over Y .*

Proof of Definition 1. By definition,

$$d_X(x_1, x_2) = \sup_{y \in Y} |d(x_1, y) - d(x_2, y)|.$$

Thus, $d_X(x_1, x_2)$ captures the worst-case deviation in how x_1 and x_2 relate to elements in Y , which establishes relational consistency. $\square$

Theorem 1 (Bipolar Lipschitz Stability). *Assume that the conditional probability $P(L \mid x)$ is Lipschitz continuous with respect to the original metric d , i.e.,*

$$|P(L \mid x) - P(L \mid x')| \leq Kd(x, x').$$

Then, under the bipolar transformation, $P(L \mid x)$ remains Lipschitz continuous with respect to d_X up to a constant factor.

Definition 2 (Bipolar Distance Operator). Let (X, Y, d) be a BMS. Define the bipolar distance operator $\mathcal{B}$ as

$$\mathcal{B}(x_i, x_j) = \sup_{y \in Y} |d(x_i, y) - d(x_j, y)|.$$

Therefore, BP-KNN generates predictions by considering the neighbor point in the bipolar space for each x neighborhood:

$$\hat{y}(x) = \text{majority}(\mathcal{N}_k(x; \mathcal{B})), \text{ where } \mathcal{N}_k \text{ neighbor-set} \quad (3)$$

In BP-K-means, centroids are also chosen in bipolar space:

$$\min_C \sum_i \mathcal{B}(x_i, \mu_{c_i})^2$$

Definition 3 (Internal Bipolar Pseudo-Metrics). Let (X, Y, d) be a BMS. The induced internal pseudo-metrics $d_X : X \times X \rightarrow \mathbb{R}^+$ and $d_Y : Y \times Y \rightarrow \mathbb{R}^+$ are defined as

$$d_X(x_1, x_2) = \sup_{y \in Y} |d(x_1, y) - d(x_2, y)|, \quad (4)$$

for all $x_1, x_2 \in X$, and

$$d_Y(y_1, y_2) = \sup_{x \in X} |d(x, y_1) - d(x, y_2)|, \quad (5)$$

for all $y_1, y_2 \in Y$. These functions define the internal pseudo-metric structure induced by (X, Y, d) [22].

The KNN model assumes that the conditional class probability $P(L | x)$ satisfies a local continuity condition. Specifically, there exists a Lipschitz constant $K > 0$ such that

$$|P(L | x) - P(L | x')| \leq K d(x, x'). \quad (6)$$

Within the BP-KNN framework, the original metric $d(x, x')$ is replaced by the internal bipolar pseudo-metric d_X . Accordingly, the distance between two samples x_i and x_j is computed as

$$d_X(x_i, x_j) = \sup_{y \in Y} |d(x_i, y) - d(x_j, y)|. \quad (7)$$

Note that d_X and d_Y satisfy the non-negativity and symmetry properties, while the triangle inequality may not necessarily hold; hence, they are referred to as pseudo-metrics.

3.1.1. Noise Suppression Under (B3) Constraint

The bipolar constraint not only defines a generalized metric structure but also induces an implicit filtering mechanism against noisy samples [22]. Consider a scenario where a point x_i within the training set is subjected to an outlier denoted by η , resulting in $\tilde{x}_i = x_i + \eta$. Within the framework of the Euclidean metric, this perturbation is directly manifested in the distance measurement: $d_{\text{standard}}(\tilde{x}_i, x_j) = d_{\text{standard}}(x_i, x_j) + \mathcal{O}(\eta)$. In the BP-KNN, the internal metric d_X is employed to apply a filter over the opposite pole Y , which constitutes the test set. The internal metric deviation for the point $\tilde{x}_i$ is denoted as Δd_X , and, as illustrated in Equations (8) and (9), is derived through the application of the inverse triangle inequality and the supremum operator.

$$\Delta d_X = \left| \sup_{y \in Y} |d(\tilde{x}_i, y) - d(x_j, y)| - \sup_{y \in Y} |d(x_i, y) - d(x_j, y)| \right| \quad (8)$$

$$\Delta d_X \leq \sup_{y \in Y} |d(\tilde{x}_i, y) - d(x_i, y)| \quad (9)$$

If the noise η compromises structural integrity (B3), then, due to constraint (B3), the value of $d(\tilde{x}_i, y)$ is constrained by the cost limit T_i from the anchor points in the noise-free region, as demonstrated by $d(\tilde{x}_i, y) > T_i \implies \tilde{w}_{ij} \rightarrow 0$ and $\tilde{w}_{ij}$ represents the weight operator, where $T_i = d(\tilde{x}_i, y_j) + d(x_j, y_j) + d(x_j, y)$. Consequently, the participation coefficient of the noisy points in the voting process approaches zero.

The variance of the KNN estimator, denoted as $\text{Var}[\hat{f}(x)]$, is contingent upon the local density of the neighboring data points. In the BP-KNN, the selection of neighbors is formulated as a solution to a constrained optimization problem in Equation (10).

$$\min \sum_{i=1}^k d_X(x_i, x_{\text{test}}) \text{ s.t. } d(x_i, y_j) \leq T_i \quad (10)$$

Let Ω_{Bipolar} be the feasible set of candidates whose distances satisfy the bipolar constraint defined by the indirect cost function T_i . The constrained set Ω_{Bipolar} is a subset of the KNN search space Ω_{Standard} ($\Omega_{\text{Bipolar}} \subseteq \Omega_{\text{Standard}}$). According to probability theory, the expected value computed under the supremum operator within a constrained sample space mitigates the influence of outliers in the Equation (11):

$$\frac{\sigma_{\text{Bipolar}}^2}{\sigma_{\text{Standard}}^2} \approx \frac{1}{|Y|} \sum_{y \in Y} \exp\left(-\frac{d(x_i, y)^2}{2\sigma^2}\right) \quad (11)$$

In scenarios where $|Y| \rightarrow \infty$ (indicating large test sets or dense anchor points), the variance of BP-KNN is asymptotically diminished in comparison to the variance of the KNN. This provides a mathematical justification for the smoother and more stable decision boundaries observed.

3.1.2. Probabilistic Interpretation of the Bipolar Constraint

The (B3) condition can be interpreted as a probabilistic filtering mechanism over the opposite pole Y . Unlike standard nearest neighbor approaches, where decisions rely on local proximity, the bipolar framework evaluates consistency across multiple anchor points.

From Equation (11), the expectation over Y can be written as

$$\mathbb{E}_{y \in Y} \left[\exp\left(-\frac{d(x_i, y)^2}{2\sigma^2}\right) \right]. \quad (12)$$

For noisy samples $\tilde{x}_i$, the expected distance satisfies

$$\mathbb{E}[d(\tilde{x}_i, y)] > \mathbb{E}[d(x_i, y)], \quad (13)$$

Which leads to:

$$\mathbb{E} \left[\exp\left(-\frac{d(\tilde{x}_i, y)^2}{2\sigma^2}\right) \right] \ll \mathbb{E} \left[\exp\left(-\frac{d(x_i, y)^2}{2\sigma^2}\right) \right] \quad (14)$$

Therefore, the expected contribution of noisy samples vanishes:

$$\mathbb{E}[w_i^{\text{noise}}] \rightarrow 0 \quad (15)$$

This shows that the bipolar constraint acts as an implicit probabilistic regularizer, suppressing geometrically inconsistent samples. Stability and generalization: For a classification

function $L(x)$, the Lipschitz constant K serves as a measure of stability. Within a bipolar structure, the $K_{Bipolar}$ is minimized in the Equation (16):

$$|L(x_i) - L(x_j)| \leq K_{Bipolar} \cdot d_X(x_i, x_j) \quad (16)$$

Given that the d_X metric assesses the distance between points in a global context (via the $Y - pole$), local discontinuities, or noise islands, are excluded from consideration by being expanded in the d_X space:

$$d_{X_{space}} : K_{Bipolar} = \inf\{K : \text{Eq. above holds}\} \implies K_{Bipolar} < K \quad (17)$$

$$d_X(x_i, x_j) \geq d(x_i, x_j) \implies K_{Bipolar} \leq K \quad (18)$$

This elucidates why the BP-KNN results in fewer misclassification errors in boundary regions. In KNN, it is assumed that the conditional class probability, exhibits local Lipschitz continuity:

$$|P(L | x) - P(L | x')| \leq Kd(x, x') \quad (19)$$

In BP-KNN, the distance $d(x, x')$ is replaced by the inner bipolar metric d_X (Definition 1):

$$d_X(x_i, x_j) = \sup_{y \in Y} |d(x_i, y) - d(x_j, y)| \quad (20)$$

Regularizing the BP-KNN distance function serves to decrease the Lipschitz constant, thereby enhancing the stability of decision-making processes.

3.1.3. Risk Reduction

The generalization error R of the KNN algorithm is constrained by the distance to the k -th nearest neighbor. In the presence of noise ϵ , the standard distance is modified to $\tilde{d} = d + \epsilon$. In the BP-KNN method, the error term is mitigated through the supremum over the opposite pole Y (test set):

$$\text{Var}(\text{BP} - \text{KNN}) = \mathbb{E} \left[\left(\sup_{y \in Y} |d(x_i, y) - d(x_{neighbor}, y)| \right)^2 \right] \quad (21)$$

By applying the (B3) condition in reverse,

$$|d(x_i, y) - d(x_j, y)| \leq d(x_i, y_a) + d(x_j, y_a) \quad (\text{subject to (B3) consistency}) \quad (22)$$

The (B3) condition acts as a geometric filter. If $x_{neighbor}$ is an outlier, $d(x_{neighbor}, y)$ may be small, but the cost T_i will exceed the threshold, forcing $\hat{w}_{ij} \rightarrow 0$. The reduction in the effective Lipschitz constant $K_{Bipolar} \leq K$ leads to a tighter risk bound:

$$R_{Bipolar} \leq R^* + \frac{K_{Bipolar}}{N^{\frac{1}{d}}} < R \quad (23)$$

where R^* is the Bayes risk. This proves that BP-KNN yields a more stable decision boundary by topologically excluding high-frequency noise from the voting process.

The above analysis establishes that the (B3) constraint introduces an implicit regularization effect. Instead of explicitly modeling noise, the bipolar structure enforces geometric consistency across poles, ensuring that inconsistent (noisy) samples are exponentially suppressed as the anchor set grows. This provides a theoretical justification for the robustness improvements observed in subsequent algorithms.

3.1.4. Computational Redundancy Elimination

The initial complexity of $\mathcal{O}((m+n)^2)$ for distance-based learning is addressed. By defining the metric on $X \times Y$, Bipolar algorithms convert the complete graph K_{m+n} into a bipartite graph $G = (X, Y, E)$. Consequently, the number of edges $|E|$ is reduced from

$$\text{Redundancy Ratio} = \frac{\frac{1}{2}(m+n)(m+n-1)}{m \cdot n} \approx 1 + \frac{m}{2n} + \frac{n}{2m} \quad (24)$$

For $n \gg m$, the reduction in complexity is asymptotic. This structural intervention eliminates $\mathcal{O}(n^2)$ intraclass computations, which possess no discriminative value in supervised learning, thereby optimizing both FLOPs and RAM utilization.

The preceding analysis demonstrates that the (B3) constraint introduces a geometric filtering mechanism that concurrently diminishes noise sensitivity and computational redundancy. Rather than altering the underlying algorithm, the bipolar structure functions as a structural regularizer, ensuring global consistency throughout the data space.

3.2. The Proposed Framework Case Study on KNN

Figure 2 illustrates the decision-making process of the proposed BP-KNN model. In standard KNN, the final class is determined solely by the direct numerical distances between the query point and all training samples. This direct distance-based mechanism can be susceptible to noisy or misleading samples that are located near the class boundaries. In contrast, BP-KNN first constructs a four-point bipolar configuration consisting of a query point q_i , secondary query anchor q_j , candidate training point p_a , and secondary training reference p_b . This geometric consistency check was implemented using a bipolar sign-space structural filter. Specifically, each candidate is assigned an implicit structural cost T_a^{sign} , and only candidates satisfying the condition $T_a^{\text{sign}} \leq \tau_i$ are retained in Ω_i . The final KNN ranking is performed among the retained candidates using only numerical distances to the query point. Therefore, BP-KNN differs from standard KNN not only in relying on direct distance but also in excluding structurally inconsistent candidate relationships and then applying numerical neighbor ranking within Ω_i .

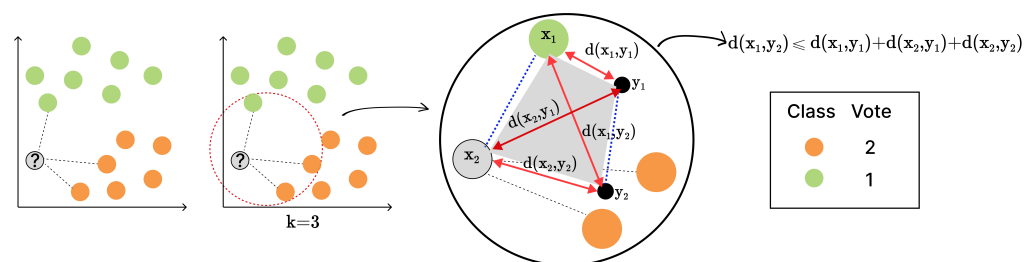


Figure 2. Distance calculation according to the BP-KNN quadrilateral inequality.

A key challenge in applying the BMS concept to supervised classification is the construction of a bipolar relational structure without confusing the bipolar poles with standard input-label notation. Therefore, demonstrated with the training set by $\mathcal{T} = \{(p_a, \ell_a)\}_{a=0}^{n-1}$, where $p_a \in \mathbb{R}^d$ is a training sample and ℓ_a is its class label. The query/test set is denoted by $\mathcal{Q} = \{q_i\}_{i=0}^{m-1}$. This notation separates class labels from the bipolar geometric construction and avoids using Y for both labels and bipolar poles.

For each query point q_i , BP-KNN constructs a four-point configuration consisting of the query point q_i , a secondary query anchor q_j , a candidate training point p_a , and a secondary training reference p_b . The secondary query anchor and the secondary training reference are selected using a deterministic cyclic two-step sliding-window rule as follows:

$$j = (i + 2) \bmod m, \quad (25)$$

$$b = (a + 2) \bmod n. \quad (26)$$

Thus, $q_j = q_{(i+2) \bmod m}$ and $p_b = p_{(a+2) \bmod n}$. The modulo operator provides circular boundary handling, so the anchor construction remains well defined for all query and training indices. For example, when $i = m - 2$, the secondary query anchor becomes q_0 , and when $i = m - 1$, it becomes q_1 .

In the operational BP-KNN procedure used in this study, the BMS-inspired structural consistency check is implemented as a bipolar sign-space filter. First, the numerical samples are standardized using the training-set statistics and then mapped into a bipolar sign representation:

$$z(u) = \text{sign}(u). \quad (27)$$

For each candidate p_a , the indirect bipolar structural cost is computed in the sign-space as

$$T_a^{\text{sign}} = \delta(z(p_a), z(q_i)) + \delta(z(p_b), z(q_i)) + \delta(z(p_b), z(q_j)). \quad (28)$$

A query-specific structural threshold is then defined as

$$\tau_i = \mu(T^{\text{sign}}) + \lambda_{\text{filter}} \sigma(T^{\text{sign}}). \quad (29)$$

where $\mu(T^{\text{sign}})$ and $\sigma(T^{\text{sign}})$ denote the mean and standard deviation of all candidate costs for query q_i , and λ_{filter} is the filtering parameter. The parameter λ_{filter} controls the permissiveness of the structural filter. Smaller values of λ_{filter} apply a tighter filter, leading to a smaller candidate set Ω_i and higher candidate exclusion rate. Larger values of λ_{filter} loosen the filter by increasing the threshold τ_i , thus holding more candidates and bringing the BP-KNN closer to the standard KNN in the limit case. Therefore, λ_{filter} regulates the balance between the structural filtering power and candidate retention; it does not change the final numerical distance metric used for ranking within Ω_i . The retained candidate set was as follows:

$$\Omega_i = \{a : T_a^{\text{sign}} \leq \tau_i\}. \quad (30)$$

The final numerical KNN ranking is performed only among the retained candidates in Ω_i . Therefore, BP-KNN separates the structural candidate filter from the final numerical neighbor ranking: the sign-space filtering identifies structurally consistent candidates, whereas the final vote is based on the numerical proximity among the retained candidates. The BP-KNN procedure is given in Algorithm 1.

Algorithm 1 BP-KNN with Bipolar Sign-Space Structural Filter**Require:** Training set $\mathcal{T} = \{(p_a, \ell_a)\}_{a=0}^{n-1}, p_a \in \mathbb{R}^d$ **Require:** Query set $\mathcal{Q} = \{q_i\}_{i=0}^{m-1}, q_i \in \mathbb{R}^d$ **Require:** Number of neighbors k **Require:** Distance metric $\delta : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$ **Require:** Sliding-window offset $s = 2$ **Require:** Filtering parameter λ_{filter} **Ensure:** Predicted label vector $\hat{\mathcal{L}} = \{\hat{\ell}_i\}_{i=0}^{m-1}$ 1: Compute the training-set mean $\mu_{\mathcal{T}}$ and standard deviation $\sigma_{\mathcal{T}}$.2: Standardize all training and query samples using $\mu_{\mathcal{T}}$ and $\sigma_{\mathcal{T}}$:

$$\tilde{p}_a = \frac{p_a - \mu_{\mathcal{T}}}{\sigma_{\mathcal{T}}}, \quad \tilde{q}_i = \frac{q_i - \mu_{\mathcal{T}}}{\sigma_{\mathcal{T}}}.$$

3: Construct the bipolar sign-space representations:

$$z(p_a) = \text{sign}(\tilde{p}_a), \quad z(q_i) = \text{sign}(\tilde{q}_i).$$

4: Initialize $\hat{\mathcal{L}} \leftarrow \emptyset$.5: **for** $i = 0$ to $m - 1$ **do**6: $j \leftarrow (i + s) \bmod m$

▷ Secondary query-anchor index

7: $q_j \leftarrow q_{(i+s) \bmod m}$

▷ Secondary query anchor.

8: Initialize $C_i \leftarrow \emptyset$

▷ List of sign-space structural costs

9: **for** $a = 0$ to $n - 1$ **do**10: $b \leftarrow (a + s) \bmod n$

▷ Secondary training reference

11: Compute the sign-space structural cost:

$$T_a^{\text{sign}} \leftarrow \delta(z(p_a), z(q_i)) + \delta(z(p_b), z(q_i)) + \delta(z(p_b), z(q_j)).$$

12: Append T_a^{sign} to C_i .13: **end for**

14: Compute the query-specific structural threshold:

$$\tau_i \leftarrow \text{mean}(C_i) + \lambda_{\text{filter}} \text{std}(C_i).$$

15: Initialize $\Omega_i \leftarrow \emptyset$.

▷ Retained candidate indices

16: **for** $a = 0$ to $n - 1$ **do**17: **if** $T_a^{\text{sign}} \leq \tau_i$ **then**18: Add a to Ω_i .19: **end if**20: **end for**21: **if** $\Omega_i = \emptyset$ **then**22: $\Omega_i \leftarrow \{0, 1, \dots, n - 1\}$.

▷ Fallback to the full candidate set

23: **end if**24: **for all** $a \in \Omega_i$ **do**

25: Compute the final numerical distance:

$$d_a \leftarrow \delta(\tilde{p}_a, \tilde{q}_i).$$

26: **end for**27: Select the k indices in Ω_i with the smallest d_a :

$$N_k^{\text{BP}}(q_i) \leftarrow \text{TopK}_{a \in \Omega_i}(d_a).$$

28: Assign the predicted label by majority voting:

$$\hat{\ell}_i \leftarrow \text{MajorityVote}\{\ell_a : a \in N_k^{\text{BP}}(q_i)\}.$$

29: Append $\hat{\ell}_i$ to $\hat{\mathcal{L}}$.30: **end for**31: **return** $\hat{\mathcal{L}}$

3.3. A Hand-Worked Example of BP-KNN

To clarify the operational mechanism of BP-KNN, this subsection provides a small hand-worked example. We use the notation $\mathcal{T} = \{(p_a, \ell_a)\}_{a=0}^{n-1}$ for the training set, where p_a is a training sample and ℓ_a is its class label. The query set is denoted by $\mathcal{Q} = \{q_i\}_{i=0}^{m-1}$. This notation avoids ambiguity between training labels and the bipolar poles used in the theoretical BMS formulation.

For each query point q_i , BP-KNN constructs a four-point configuration consisting of the candidate training point p_a , the secondary training reference p_b , the query point q_i , and the secondary query anchor q_j . In the implementation used in this study, the secondary anchors are selected by a deterministic cyclic two-step sliding-window rule:

$$j = (i + 2) \bmod m, \quad (31)$$

$$b = (a + 2) \bmod n. \quad (32)$$

Therefore,

$$q_j = q_{(i+2) \bmod m}, \quad p_b = p_{(a+2) \bmod n}. \quad (33)$$

The modulo operation provides circular boundary handling. The same cyclic rule is applied to the training samples. Thus, the anchor construction is deterministic, reproducible, and well-defined for all query and training indices.

In the formulation of BP-KNN, the standardized numerical samples are first mapped into a bipolar sign-space:

$$z(u) = \text{sign}(u). \quad (34)$$

For each candidate p_a , the BMS-inspired indirect structural cost is computed in the bipolar sign-space as

$$T_a^{\text{sign}} = \delta(z(p_a), z(q_i)) + \delta(z(p_b), z(q_i)) + \delta(z(p_b), z(q_j)). \quad (35)$$

This cost corresponds to the three bounding path terms in the quadrilateral configuration. A query-specific structural threshold is then defined as

$$\tau_i = \mu(T^{\text{sign}}) + \lambda\sigma(T^{\text{sign}}), \quad (36)$$

where $\mu(T^{\text{sign}})$ and $\sigma(T^{\text{sign}})$ are the mean and standard deviation of the candidate costs for query q_i , and λ_{filter} is the filter parameter. The retained candidate set is

$$\Omega_i = \{a : T_a^{\text{sign}} \leq \tau_i\}. \quad (37)$$

The final numerical neighbor ranking is then performed only among the candidates in Ω_i :

$$N_k^{\text{BP}}(q_i) = \text{TopK}_{a \in \Omega_i} \delta(p_a, q_i). \quad (38)$$

Finally, the predicted class is obtained by majority voting over the labels of the selected neighbors.

Consider a two-dimensional benchmark dataset with $k = 3$, $\lambda_{\text{filter}} = 1$, and Euclidean distance. Assume that all coordinates have already been standardized. Let the target query point be

$$q_i = (0.10, 0.10), \quad (39)$$

and let the secondary query anchor selected by Equation (31) be

$$q_j = (-0.20, -0.20). \quad (40)$$

The training samples are given in Table 3.

Table 3. Benchmark dataset training samples used in the hand-worked BP-KNN example.

Index a	Training Point p_a	Label ℓ_a	Secondary Reference $p_b = p_{(a+2) \bmod 6}$
0	(0.15, 0.12)	A	$p_2 = (2.00, 2.00)$
1	(1.30, 1.25)	A	$p_3 = (2.20, 2.10)$
2	(2.00, 2.00)	A	$p_4 = (-0.05, -0.04)$
3	(2.20, 2.10)	A	$p_5 = (-0.08, -0.06)$
4	(−0.05, −0.04)	B	$p_0 = (0.15, 0.12)$
5	(−0.08, −0.06)	B	$p_1 = (1.30, 1.25)$

The sign-space representations of the query and anchor are

$$z(q_i) = (+, +), \quad z(q_j) = (-, -). \quad (41)$$

For p_0, p_1, p_2, p_3 , the sign-space representation is $(+, +)$. For p_4, p_5 , the sign-space representation is $(-, -)$. Therefore, the Euclidean distance between two identical polarity vectors is 0, whereas the distance between $(+, +)$ and $(-, -)$ is

$$\sqrt{(1 - (-1))^2 + (1 - (-1))^2} = \sqrt{8} = 2.828. \quad (42)$$

The resulting sign-space structural costs are shown in Table 4.

Table 4. Computation of the bipolar sign-space structural cost T_a^{sign} .

a	Label	$z(p_a)$	$z(p_b)$	T_a^{sign}
0	A	(+, +)	(+, +)	$0 + 0 + 2.828 = 2.828$
1	A	(+, +)	(+, +)	$0 + 0 + 2.828 = 2.828$
2	A	(+, +)	(−, −)	$0 + 2.828 + 0 = 2.828$
3	A	(+, +)	(−, −)	$0 + 2.828 + 0 = 2.828$
4	B	(−, −)	(+, +)	$2.828 + 0 + 2.828 = 5.656$
5	B	(−, −)	(+, +)	$2.828 + 0 + 2.828 = 5.656$

The mean and standard deviation of the six structural costs are

$$\mu(T^{\text{sign}}) = 3.771, \quad (43)$$

$$\sigma(T^{\text{sign}}) = 1.333. \quad (44)$$

With $\lambda_{\text{filter}} = 1$, the structural threshold is

$$\tau_i = 3.771 + 1(1.333) = 5.104. \quad (45)$$

Hence, the retained candidate set is

$$\Omega_i = \{0, 1, 2, 3\}. \quad (46)$$

Candidates p_4 and p_5 are excluded because

$$T_4^{\text{sign}} = T_5^{\text{sign}} = 5.656 > \tau_i = 5.104. \quad (47)$$

These two candidates are numerically close to q_i , but their polarity profiles are inconsistent with the query-anchor configuration. Therefore, they are removed by the bipolar sign-space structural filter before the final numerical ranking.

Standard KNN ranks all training samples by their direct numerical distances to q_i , as shown in Table 5.

Table 5. Standard KNN direct numerical distances to the query point q_i .

Candidate	Label	Numerical Distance to q_i
p_0	A	0.054
p_4	B	0.205
p_5	B	0.241
p_1	A	1.662
p_2	A	2.687
p_3	A	2.900

For $k = 3$, standard KNN selects

$$\{p_0(A), p_4(B), p_5(B)\}, \quad (48)$$

and therefore predicts

$$\hat{\ell}_{\text{KNN}} = B. \quad (49)$$

In contrast, BP-KNN first constructs

$$\Omega_i = \{p_0, p_1, p_2, p_3\}. \quad (50)$$

The final numerical ranking is then performed only inside Ω_i . The three nearest retained candidates are

$$\{p_0(A), p_1(A), p_2(A)\}. \quad (51)$$

Therefore, BP-KNN predicts

$$\hat{\ell}_{\text{BP-KNN}} = A. \quad (52)$$

This example demonstrates how BP-KNN differs from standard KNN. Standard KNN relies only on direct numerical proximity, whereas BP-KNN first evaluates whether each candidate participates in a structurally consistent bipolar relation. As a result, numerically close but polarity-inconsistent candidates may be excluded before the final voting stage.

In this example, standard KNN computes six full numerical distances and ranks all six candidates. BP-KNN computes the sign-space structural costs for six candidates, excludes two candidates, and computes final numerical distances only for the four retained candidates. Thus,

$$|\Omega_i| = 4, \quad (53)$$

$$\text{candidate exclusion rate} = \frac{2}{6} = 33.33\%. \quad (54)$$

The number of final numerical distance evaluations is reduced from 6 to 4. This example separates the inexpensive sign-space structural evaluation from the final numerical distance ranking and illustrates, shown in the Table 6, how the proposed filter reduces the candidate set before the final KNN vote.

Table 6. Operation-count comparison for the hand-worked example.

Method	Final Numerical Distance Evaluations	Candidate Exclusion Rate
Standard KNN	6	0%
BP-KNN	$ \Omega_i = 4$	$\frac{2}{6} \times 100 = 33.33\%$

3.4. The Proposed Framework Case Study on K-Means

Standard K-means, is an iterative fixed-point search method that minimizes the objective function by alternating between assignment and update steps [27]. In addition to the standard structure, the BP-K-means algorithm is based on the concept of BMS. In BP-K-means, the BMS intervention reformulates this process into a geometrically constrained optimization problem, as defined in Definition 1. This approach uses a distance function defined between the dataset and centroid. The most important element of the bipolar structure is inequality, as defined in condition (B3). In the standard K-means approach, the assignment weight w_{ij} for point x_i to centroid c_j is determined solely by local proximity. The primary objective is to minimize the quantization error $\mathcal{J}$ across the entire space $\mathbb{R}^d$.

$$\mathcal{J} = \sum_{i=1}^n \sum_{j=1}^k w_{ij} \|x_i - c_j\|^2, \quad w_{ij} \in \{0, 1\} \quad (55)$$

In BP-K-means, the bipolar assignment score is used as a constrained replacement for the standard K-means assignment rule. For each data point x_i and candidate centroid $c_j^{(t)}$, two cyclic reference indices are selected as

$$b = ((i + 1) \bmod n) + 1, \quad \ell = ((j + 1) \bmod K) + 1, \quad (56)$$

where x_b denotes the secondary data-point anchor and $c_\ell^{(t)}$ denotes the secondary centroid anchor. The BMS-inspired indirect assignment threshold is then defined as

$$T_{ij}^{(t)} = \delta(x_i, c_\ell^{(t)}) + \delta(x_b, c_\ell^{(t)}) + \delta(x_b, c_j^{(t)}). \quad (57)$$

Accordingly, the constrained assignment score is computed as

$$s_{ij}^{(t)} = \begin{cases} \delta(x_i, c_j^{(t)}), & \text{if } \delta(x_i, c_j^{(t)}) \leq T_{ij}^{(t)}, \\ \infty, & \text{otherwise.} \end{cases} \quad (58)$$

The cluster assignment is then obtained as

$$a^{(t+1)}(i) = \arg \min_{1 \leq j \leq K} s_{ij}^{(t)}. \quad (59)$$

Equivalently, the binary assignment weight can be written as

$$\hat{w}_{ij}^{(t+1)} = \begin{cases} 1, & \text{if } j = a^{(t+1)}(i), \\ 0, & \text{otherwise.} \end{cases} \quad (60)$$

The feasible bipolar assignment set at iteration t is therefore defined as

$$\Omega_{\text{Bipolar}}^{(t)} = \{(x_i, c_j^{(t)}) \in X \times C^{(t)} : \delta(x_i, c_j^{(t)}) \leq T_{ij}^{(t)}\}. \quad (61)$$

This formulation ensures that a data point is assigned to a centroid only when the direct point–centroid assignment is geometrically consistent with the BMS-inspired indirect path. If $\delta(x_i, c_j^{(t)}) > T_{ij}^{(t)}$, the assignment is treated as infeasible and excluded from the centroid update. This hard feasibility mechanism stabilizes centroid updates by reducing geometrically inconsistent point–centroid assignments, thereby helping the algorithm avoid boundary oscillations commonly observed in standard K-means.

This hard filtering mechanism makes the algorithm compact in terms of distance and generates cluster structures based on the geometric consistency defined by the BMS. This leads to more stable centroid updates and, as our experiments show, faster convergence by avoiding the clustering oscillations often seen in standard K-means. This strict filtering prevents a data point lying between two clusters from being repeatedly reassigned, thus forcing the algorithm to determine more stable cluster boundaries at an earlier stage. The c_b functions as an intermediary. If the direct distance between point x_i and center c_j exceeds this indirect intermediary path (i.e., T_i), the bipolar rule deems this assignment geometrically inconsistent and consequently prohibits it.

$$\Omega_{\text{Bipolar}}^{(t)} = \{(x_i, c_j) \in X \times C : d(x_i, c_j) \leq T_j\}. \quad (62)$$

Equation (62) delineates a constrained feasible set, denoted as $\Omega_{\text{Standard}}^{(t)}$, for each iteration t . Given that $(\Omega_{\text{Bipolar}})$ is a subset of $(\Omega_{\text{Standard}})$, the algorithm efficiently reduces the search space by assigning an infinite cost to assignments that compromise the geometric integrity of the bipolar space.

Convergence acceleration through contraction mapping: The K-means update can be conceptualized as a mapping $f : C \rightarrow C$.

Theorem 2. Consider a complete BMS denoted as (X, Y, d) , and let there be a contraction mapping $f : (X, Y, d) \Rightarrow (X, Y, d)$. It follows that the function $f : X \cup Y \rightarrow X \cup Y$ possesses a unique fixed point [22].

In accordance with the Theorem 2, the sequence of centroids $c^{(t)}$ converges to a unique fixed point u . The error at step $t + 1$ is as follows:

$$\|c^{(t+1)} - u\| \leq \lambda_{\text{Bipolar}} \|c^{(t)} - u\|, \quad (63)$$

$$\lambda_{\text{Bipolar}} = \sup_{c \in \Omega_{\text{Bipolar}}} \|I - \nabla^2 J(c)\|. \quad (64)$$

The acceleration is demonstrated through the analysis of the Contraction Coefficient λ . In the classic K-means algorithm, $\lambda_{\text{Standard}}$ is affected by boundary points characterized by high ambiguity, specifically those equidistant to multiple centroids, which results in slow and oscillatory convergence. In contrast, BP-K-means employs the hard constraint T_i as a topological regularize. The (B3) condition effectively filters out unstable boundary assignments, where the gradient $\nabla \mathcal{J}$ approaches zero and the Hessian $\nabla^2 \mathcal{J}$ is ill-conditioned, thereby ensuring that the mapping remains within a region of high curvature. This leads to $\lambda_{\text{Bipolar}} < \lambda_{\text{Standard}}$.

Iteration complexity reduction: The number of iterations T necessary to achieve a stability tolerance ϵ exhibits a logarithmic relationship with respect to λ , and $T \approx \frac{\ln(\epsilon)}{\ln(\lambda)}$. Given $\lambda_{\text{Bipolar}} < \lambda_{\text{Standard}}$, it can be deduced that:

$$T_{\text{Bipolar}} = \frac{\ln(\epsilon)}{\ln(\lambda_{\text{Bipolar}})} < T_{\text{Standard}} = \frac{\ln(\epsilon)}{\ln(\lambda_{\text{Standard}})}. \quad (65)$$

This formalizes the empirical observation that BP-K-means converges in significantly fewer steps compared to the standard model. The BMS framework imposes a geometric momentum that prevents the centroids from remaining in low-information regions of the data manifold, thereby ensuring a more stable trajectory toward a fixed point or locally consistent solution.

The BP-K-means is formalized in Algorithm 2. The algorithm modifies the standard Lloyd procedure during the Assignment Step (Lines 12–24). For each data point x_i and each potential center point $c_j^{(t)}$, a bipolar reference pair $(x_b, c_\ell^{(t)})$ is determined (Lines 13–15). The assignment is treated as a constrained optimization. If the direct distance indirect path cost exceeds T_{ij} , the assignment is penalized with an infinite cost (Lines 16–21). This strict constraint ensures that the centerpoints are updated only using structurally coherent sets (Lines 25–32). Consequently, the sequence of centerpoints converges to a stable fixed point in significantly fewer iterations, as the gatekeeper (B3) prevents the centerpoints from converging to unstable class boundaries.

Algorithm 2 The pseudocode of BP-K-means

Require: Dataset $D = \{x_i\}_{i=1}^n$, number of clusters K , maximum number of iterations $T_{\max}$, tolerance ε

Ensure: Cluster assignments $a(i)$ and centroids $C = \{c_j\}_{j=1}^K$

```

1: Initialize centroids  $C^{(0)} = \{c_j^{(0)}\}_{j=1}^K$ 
2:  $t \leftarrow 0$ 
3: repeat
4:   Store previous centroids:  $C_{\text{old}} \leftarrow C^{(t)}$ 
5:   for  $i = 1, \dots, n$  do
6:     Select the secondary data-point anchor index:

$$b \leftarrow ((i + 1) \bmod n) + 1$$

7:     Let  $x_b$  be the data point at index  $b$ 
8:     for  $j = 1, \dots, K$  do
9:       Select the secondary centroid anchor index:

$$\ell \leftarrow ((j + 1) \bmod K) + 1$$

10:      Let  $c_\ell^{(t)}$  be the centroid at index  $\ell$ 
11:      Compute the BMS-inspired indirect assignment threshold:

$$T_{ij}^{(t)} \leftarrow \delta(x_i, c_\ell^{(t)}) + \delta(x_b, c_\ell^{(t)}) + \delta(x_b, c_j^{(t)})$$

12:      if  $\delta(x_i, c_j^{(t)}) \leq T_{ij}^{(t)}$  then
13:         $s_{ij}^{(t)} \leftarrow \delta(x_i, c_j^{(t)})$ 
14:      else
15:         $s_{ij}^{(t)} \leftarrow \infty$ 
16:      end if
17:    end for
18:    Assign  $x_i$  to the feasible centroid with the minimum constrained assignment score:

$$a^{(t+1)}(i) \leftarrow \arg \min_{1 \leq j \leq K} s_{ij}^{(t)}$$

19:    if  $s_{i, a^{(t+1)}(i)}^{(t)} = \infty$  then
20:      Assign  $x_i$  to the nearest centroid using the standard fallback rule:

$$a^{(t+1)}(i) \leftarrow \arg \min_{1 \leq j \leq K} \delta(x_i, c_j^{(t)})$$

21:    end if
22:  end for
23:  for  $j = 1, \dots, K$  do
24:    Update centroid:

$$c_j^{(t+1)} \leftarrow \frac{1}{|\{i : a^{(t+1)}(i) = j\}|} \sum_{i: a^{(t+1)}(i) = j} x_i$$

25:  end for
26:   $t \leftarrow t + 1$ 
27: until  $\|C^{(t)} - C_{\text{old}}\| < \varepsilon$  or  $t = T_{\max}$ 
28: return  $a(i)$  and  $C^{(t)}$ 

```

4. Experimental Study

This section presents the experimental setup and performance analyses performed on six datasets to evaluate the effectiveness of the bipolar models. All experiments were conducted in Python on a computer equipped with an Intel® Core™ Ultra 9 185H processor (2.50 GHz) x64, 32 GB DDR5 RAM (6400 MT/s), Intel® Arc™ Graphics (128 MB VRAM), and 2 TB SSD storage. Furthermore, all experiments used the same system infrastructure to eliminate hardware-related variance, and the performance comparisons were conducted under equal conditions.

To assess the efficacy of the bipolar filtering operator, it was applied to standard distance-based algorithms, including KNN and K-means. These baseline models served as reference points, whereas their bipolar counterparts illustrated the impact of structural filter. The observed differences are not due to replacing the base learner with a different model family, but to adding the BMS-based structural filtering constraint to the same underlying distance-based algorithm. This indicates that the proposed enhancement is independent of the underlying algorithm and is driven instead by the geometric consistency enforced by the (B_3) constraint. By eliminating the (B_3) constraint, the model reverts to conventional KNN, thereby allowing a direct comparison between KNN and BP-KNN. Accordingly, this comparison should be interpreted as a controlled case-study evaluation that isolates the effect of the BMS-based structural filtering operator while keeping the underlying base algorithm unchanged, rather than as a complete benchmark against all advanced distance-based methods.

The BP-KNN and BP-K-means methodologies were subjected to a comprehensive evaluation using six distinct datasets, each exhibiting variations in dimensions, domains, and sample sizes, to substantiate their robustness across diverse scenarios. Initial validation used standard, low-dimensional benchmarks from the UCI repository—Iris, Breast Cancer, and Wine—chosen for their balanced and well-established benchmark properties. To assess performance on more complex, real-world problems, we employed the CDC Diabetes [28], Abalone [29], and MWS [30] datasets. Abalone tested the handling of mixed data types and unbalanced distributions, while MWS evaluated scalability on millions of observations. This tiered evaluation framework, from small benchmarks to large-scale applications, demonstrates the model's generalizability, scalability, and domain-agnostic robustness, as detailed in Table 7.

Table 7. Basic properties of the datasets used in this study.

Dataset	# of Samples / # of Features	Type	Balanced or Unbalanced	Data Type
Iris	150/4	Multi-class (3 classes)	Balanced	Numeric
Breast Cancer	569/30	Binary	Balanced	Numeric
Wine	178/13	Multi-class (3 classes)	Balanced	Numeric
Abalone	4177/8	Multi-class (28 classes)	Balanced	Mixed
CDC Diabetes	253,680/21	Binary	Unbalanced (6:1)	Mixed
MWS	2,787,159/27	Binary	Unbalanced (2:1)	Numeric

Therefore, the comparison with the standard KNN and K-means aims to isolate the effect of the BMS-based structural filter while keeping the underlying algorithm constant. Advanced methods such as OTI/EOTI [7], MCML/BMCML [12], and NNS-DML [9] address related but different design goals, including fast searching, metric learning, and dynamic neighborhood selection. Therefore, these methods are considered separately in terms of their mechanisms, computational balance, and scalability. To strengthen this positioning, this study also reports rigorous validation-based parameter selection, permutation sensitivity analysis, candidate reduction statistics, and FLOP-level operation count analysis for the BP-KNN.

4.1. Comparative Performance Results for the Case Study KNN

The performance of BP-KNN was evaluated in comparison to the standard KNN across six distinct datasets, encompassing both benchmark and real-world data. The findings, as presented in Tables 8 and 9, indicate that the incorporation of constraint ($B3$) enhances stability and efficiency by altering the classification behavior.

Table 8. Performance comparisons of BP-KNN and standard KNN on different datasets.

Model	Dataset	Distance Metric	Accuracy	F1-Macro	F1-Weighted	ROC
KNN	Wine	Euclidean	0.8055	0.8025	0.8066	0.8540
		Manhattan	0.8333	0.8286	0.8333	0.8723
		Chebyshev	0.75	0.7386	0.7477	0.8055
		Minkowski	0.8055	0.8025	0.8066	0.8540
BP-KNN	Wine	Euclidean	0.8888	0.8881	0.8897	0.9208
		Manhattan	0.8888	0.8857	0.8888	0.9149
		Chebyshev	0.8611	0.8609	0.8618	0.9025
		Minkowski	0.8888	0.8881	0.8897	0.9208
KNN	Breast Cancer	Euclidean	0.9123	0.9013	0.9099	0.8860
		Manhattan	0.9122	0.8999	0.9092	0.8809
		Chebyshev	0.9122	0.9013	0.9099	0.8859
		Minkowski	0.9123	0.9013	0.9099	0.8860
BP-KNN	Breast Cancer	Euclidean	0.9210	0.9117	0.9192	0.8978
		Manhattan	0.9210	0.9128	0.9198	0.9027
		Chebyshev	0.9210	0.9117	0.9192	0.8978
		Minkowski	0.9210	0.9117	0.9192	0.8978
KNN	Iris	Euclidean	0.9333	0.9333	0.9333	0.9500
		Manhattan	0.9333	0.9333	0.9333	0.9499
		Chebyshev	1.0	1.0	1.0	1.0
		Minkowski	0.9333	0.9333	0.9333	0.9500
BP-KNN	Iris	Euclidean	0.9333	0.9327	0.9327	0.9500
		Manhattan	0.9333	0.9327	0.9327	0.9500
		Chebyshev	0.9666	0.9666	0.9666	0.9750
		Minkowski	0.9333	0.9327	0.9327	0.9500
KNN	CDC Diabetes	Euclidean	0.8599	0.5469	0.8186	0.6793
		Manhattan	0.8607	0.5523	0.8203	0.6861
		Chebyshev	0.8591	0.5267	0.8128	0.6667
		Minkowski	0.8599	0.5469	0.8186	0.6793
BP-KNN	CDC Diabetes	Euclidean	0.8625	0.5675	0.8251	0.7005
		Manhattan	0.8647	0.5677	0.8262	0.7184
		Chebyshev	0.8593	0.5461	0.8181	0.6737
		Minkowski	0.8625	0.5675	0.8251	0.7005
KNN	Abalone	Euclidean	0.2607	0.1044	0.2416	0.5286
		Manhattan	0.2580	0.0989	0.2381	0.5204
		Chebyshev	0.2592	0.0987	0.2403	0.5312
		Minkowski	0.2607	0.1044	0.2416	0.5286
BP-KNN	Abalone	Euclidean	0.2696	0.0999	0.2460	0.7429
		Manhattan	0.2886	0.1220	0.2696	0.7451
		Chebyshev	0.2658	0.0907	0.2400	0.7589
		Minkowski	0.2696	0.0999	0.2460	0.7429
KNN	MWS	Euclidean	0.6375	0.3893	0.4964	0.5000
		Manhattan	0.6298	0.3796	0.4820	0.5019
		Chebyshev	0.6352	0.3903	0.4967	0.5093
		Minkowski	0.6375	0.3893	0.4964	0.5000
BP-KNN	MWS	Euclidean	0.6612	0.4802	0.5701	0.7228
		Manhattan	0.6593	0.4783	0.5674	0.7185
		Chebyshev	0.6599	0.4790	0.4686	0.7196
		Minkowski	0.6612	0.4802	0.5701	0.7228

Table 9. RAM and time consumptions of BP-KNN and standard KNN on different datasets.

Model	Dataset	Distance Metric	RAM Consumption (MB)	Time Consumption (s)
KNN	Wine	Euclidean	197.75	0.02048
		Manhattan	197.48	0.0145
		Chebyshev	197.71	0.0156
		Minkowski	197.75	0.0198
BP-KNN	Wine	Euclidean	195.24	0.00199
		Manhattan	195.46	0.00200
		Chebyshev	195.88	0.00201
		Minkowski	195.24	0.00199
KNN	Breast Cancer	Euclidean	199.22	0.13870
		Manhattan	199.10	0.12216
		Chebyshev	199.25	0.13568
		Minkowski	199.22	0.13870
BP-KNN	Breast Cancer	Euclidean	198.63	0.0167
		Manhattan	198.45	0.0130
		Chebyshev	198.50	0.0169
		Minkowski	198.63	0.0167
KNN	Iris	Euclidean	198.59	0.0125
		Manhattan	198.70	0.0156
		Chebyshev	199.85	0.0119
		Minkowski	198.59	0.0125
BP-KNN	Iris	Euclidean	197.62	0.0009
		Manhattan	197.87	0.0009
		Chebyshev	198.10	0.0009
		Minkowski	197.62	0.0009
KNN	CDC Diabetes	Euclidean	384.93	125,794.345
		Manhattan	384.22	84,994.606
		Chebyshev	385.25	97,279.042
		Minkowski	384.93	125,794.345
BP-KNN	CDC Diabetes	Euclidean	252.88	29,273.815
		Manhattan	250.92	20,463.997
		Chebyshev	248.47	22,511.460
		Minkowski	252.88	29,273.815
KNN	Abalone	Euclidean	184.79	54.7245
		Manhattan	185.09	34.9358
		Chebyshev	184.34	35.2408
		Minkowski	184.79	54.7245
BP-KNN	Abalone	Euclidean	181.86	4.8141
		Manhattan	184.96	25.5331
		Chebyshev	181.34	23.1008
		Minkowski	181.86	4.8141
KNN	MWS	Euclidean	2713.28	155,625.9605
		Manhattan	2713.30	155,817.9605
		Chebyshev	2706.61	152,513.9605
		Minkowski	2713.28	155,625.9605
BP-KNN	MWS	Euclidean	689.47	13,425.1468
		Manhattan	688.94	17,107.3488
		Chebyshev	689.23	16,976.3488
		Minkowski	689.47	13,425.1468

According to the results in Table 8, BP-KNN exhibits greater resistance to local noise compared to the standard KNN. In high-dimensional spaces, such as MWS, and overlapping datasets, such as Abalone, BP-KNN significantly enhanced the F1-Macro scores (e.g., from 0.3893 to 0.4802 in MWS). This improvement illustrates that the bipolar metric effectively prunes geometrically inconsistent neighbors that the standard KNN fails to identify. In the unbalanced CDC Diabetes dataset, the geometric consistency cost $T_{Bipolar}$ yielded higher ROC AUC values by better preserving the minority class clusters. Moreover, a specific theoretical phenomenon, termed the regulation effect, was observed in the Iris Dataset. While the standard KNN achieved an accuracy of 1.0 with the Chebyshev metric, BP-KNN attained an accuracy of 0.9666. This slight reduction is interpreted as a success in mitigating overfitting rather than a shortcoming. By reducing the effective Lipschitz constant ($K_{Bipolar} < K_{Standard}$), the BMS framework avoids forming irregular boundaries around isolated noise points, resulting in a more geometrically generalized and robust decision manifold. This effect was further validated in the Wine dataset, where BP-KNN outperformed the standard model significantly.

A significant reduction in computational load is evident from the results in Table 9. On the large-scale MWS dataset, BP-KNN demonstrated an 11.6-fold increase in processing speed and a 75% reduction in RAM usage. This outcome supports the computational interpretation of the proposed BP-KNN mechanism. The bipolar sign-space structural filter reduces the candidate set before the final numerical ranking, thereby replacing a portion of exhaustive full-distance evaluations with the 4-FLOP structural validation.

4.2. Strict Validation-Based Parameter Selection

In the main experiments, KNN-based models were initially evaluated under a constant neighborhood setting of $k = 5$ common to all distance measures. This constant k protocol was used to isolate the effect of the proposed bipolar structural filter without changing the underlying KNN configuration. However, because the optimum neighborhood size can vary between datasets, a rigorous validation-based hyperparameter analysis was performed for both standard KNN and BP-KNN. This additional analysis was designed to verify whether the comparative behavior of BP-KNN remains stable when k is selected from a validation set, rather than changing the constant k in Tables 8 and 9.

For the standard KNN, k was selected from a predefined candidate grid using the validation macro F1. For BP-KNN, both the neighborhood dimension k and the bipolar sign-space structural filter parameter λ_{filter} were selected using the same validation protocol. In this analysis, λ_{filter} was treated as a validation-selected filtering hyperparameter that checked the precision of the conserved candidate set Ω_i , and it was selected from the candidate grid $\{0.5, 1.0, 1.5, 2.0\}$.

Table 10 confirms that the selected k values vary between the datasets and methods, and that a single neighborhood dimension is not universally optimal. However, while the results adjusted with validation remain generally consistent with Table 8, BP-KNN improves on standard KNN across various datasets, including Wine, Breast Cancer, Abalone, CDC Diabetes, and MWS. These results demonstrate that the observed behavior of BP-KNN is not solely a consequence of the original $k = 5$ setting.

Table 10. Additional strict validation-based hyperparameter check for standard KNN and BP-KNN.

Dataset	Method	k	λ_{filter}	Test Accuracy	Test Macro-F1	Test Weighted-F1	Test ROC-AUC
Iris	Standard KNN	7	–	1.0000	1.0000	1.0000	1.0000
Iris	BP-KNN	7	1.0	0.9661	0.9687	0.9678	0.9756
Wine	Standard KNN	7	–	0.7956	0.7928	0.7983	0.8409
Wine	BP-KNN	5	0.5	0.8888	0.8881	0.8897	0.9208
Breast Cancer	Standard KNN	7	–	0.9474	0.9422	0.9468	0.9335
Breast Cancer	BP-KNN	3	0.5	0.9474	0.9429	0.9471	0.9385
Abalone	Standard KNN	15	–	0.6005	0.6028	0.6046	0.7012
Abalone	BP-KNN	21	0.5	0.6359	0.6387	0.6402	0.7193
CDC Diabetes	Standard KNN	3	–	0.8317	0.5849	0.8158	0.5744
CDC Diabetes	BP-KNN	3	0.5	0.8333	0.5887	0.8201	0.5969
MWS	Standard KNN	3	–	0.5520	0.5443	0.5281	0.6353
MWS	BP-KNN	3	1.0	0.5760	0.5737	0.5650	0.6447

4.2.1. Permutation-Sensitivity Analysis

Because the secondary query anchor in BP-KNN is selected by the deterministic cyclic rule

$$q_j = q_{(i+2) \bmod m},$$

the anchor assigned to each query sample depends on the ordering of the query set. Therefore, BP-KNN is not strictly permutation-invariant with respect to the test-set order. To examine whether this order dependence affects the reported predictive performance, we conducted a permutation-sensitivity experiment.

For each dataset, the training set was kept fixed, while the query/test set was randomly permuted R times. For each permutation, BP-KNN was re-evaluated using the same k , λ_{filter} , distance metric, preprocessing procedure, and deterministic cyclic two-step anchor rule. The corresponding true labels were permuted together with the test samples to preserve the correct sample–label correspondence. We then reported the mean and standard deviation of accuracy, macro-F1, weighted-F1, candidate-exclusion rate, and average retained candidate-set size across the completed permutations.

The results in Table 11 show that the variation across random test-set permutations remains small. Although the cyclic anchor rule is order-dependent by construction, the predictive metrics and retained candidate-set sizes are stable across permutations. This indicates that BP-KNN is empirically robust to query-order perturbations. The stability is mainly due to the query-specific structural threshold τ_i , which evaluates the distribution of sign-space structural costs over the candidate set for each query. As a result, local changes in the secondary query anchor do not substantially alter the retained candidate structure or the final majority vote.

Table 11. Permutation-sensitivity analysis of BP-KNN under random test-set permutations.

Dataset	R	Accuracy	Macro-F1	Weighted-F1	Candidate Exclusion Rate	Avg. $ \Omega_i $
Iris	30	0.9333 ± 0.0000	0.9327 ± 0.0000	0.9327 ± 0.0000	$19.78 \pm 0.62\%$	96.26 ± 0.75
Wine	30	0.9722 ± 0.0000	0.9718 ± 0.0000	0.9724 ± 0.0000	$16.79 \pm 0.30\%$	118.17 ± 0.42
Breast Cancer	30	0.9564 ± 0.0016	0.9524 ± 0.0017	0.9561 ± 0.0016	$18.22 \pm 0.17\%$	372.12 ± 0.79
Abalone	30	0.5969 ± 0.0002	0.5971 ± 0.0002	0.5996 ± 0.0002	$20.32 \pm 0.08\%$	2662.11 ± 2.63
CDC Diabetes	30	0.8476 ± 0.0001	0.5985 ± 0.0002	0.8266 ± 0.0001	$16.03 \pm 0.01\%$	$170,420.69 \pm 7.33$
MWS	30	0.6477 ± 0.0051	0.5282 ± 0.0066	0.5934 ± 0.0057	$20.16 \pm 0.05\%$	$39,918.83 \pm 22.87$

Nevertheless, this empirical robustness should be interpreted within the scope of the tested datasets and the random permutation protocol. Because the secondary query anchor is selected by a deterministic cyclic rule, BP-KNN may be less reliable when the query/test set ordering produces structurally unrelated anchor pairs. Such a situation may occur when the test set is formed by concatenating heterogeneous acquisition batches, distribution-shifted subsets, strongly imbalanced class blocks, or adversarially arranged samples. In these cases, a query point may be paired with a secondary anchor from a different local data regime, which can distort the sign-space structural cost T_a^{sign} , alter the thresholded retained candidate set Ω_i , and potentially affect the final majority voting. Therefore, the current cyclic anchor rule is most appropriate when the query set is randomly ordered or when neighboring query indices do not encode a strong batch, temporal, or distributional structure. For datasets with meaningful ordering or heterogeneous acquisition blocks, additional care is required, such as randomizing the query order, using batch-aware shuffling, selecting anchors from local neighborhoods, or aggregating multiple secondary anchors. We consider these anchor-selection strategies to be an important direction for future research.

4.2.2. Operation-Count and Candidate-Reduction Analysis

To complement the wall-clock runtime and memory measurements, we added a direct operation-count analysis for BP-KNN. For each dataset, standard KNN requires N full numerical distance evaluations per query instance, where N is the number of training candidates. In contrast, BP-KNN first applies the bipolar sign-space structural filter and then computes final full numerical distances only for the retained candidate set Ω_i . Therefore, the number of final full numerical distance evaluations is reduced from N to $|\Omega_i|$ per query instance.

The results in Table 12 directly quantify the computational effect of the bipolar sign-space structural filter. Standard KNN computes full numerical distances for all training candidates, whereas BP-KNN reduces the final full-distance ranking workload by retaining only Ω_i . For example, on the MWS diagnostic subset, standard KNN requires 250,000,000 full numerical distance evaluations, whereas BP-KNN requires approximately 199,594,155, corresponding to an average candidate-exclusion rate of 20.16%. Similarly, on CDC Diabetes, BP-KNN reduces the final full-distance evaluations from 10.30 billion to approximately 8.65 billion. These results show that the proposed filter reduces the final numerical ranking workload rather than merely changing the voting rule after a complete exhaustive ranking.

Table 12. Operation-count and candidate-reduction analysis of BP-KNN compared with standard KNN.

Dataset	N	m	KNN Full Distances	BP-KNN Full Distances	Avg. Filtered Candidates	Avg. $ \Omega_i $	Exclusion Rate
Iris	120	30	3600	2888	23.74	96.26 ± 0.75	$19.78 \pm 0.62\%$
Wine	142	36	5112	4254	23.83	118.17 ± 0.42	$16.79 \pm 0.30\%$
Breast Cancer	455	114	51,870	42,421	82.88	372.12 ± 0.79	$18.22 \pm 0.17\%$
Abalone	3341	836	2,793,076	2,225,526	678.89	2662.11 ± 2.63	$20.32 \pm 0.08\%$
CDC Diabetes	202,944	50,736	10,296,566,784	8,646,464,038	32,523.31	$170,420.69 \pm 7.33$	$16.03 \pm 0.01\%$
MWS ^a	50,000	5000	250,000,000	199,594,155	10,081.17	$39,918.83 \pm 22.87$	$20.16 \pm 0.05\%$

Note. N denotes the number of training candidates and m denotes the number of evaluated query/test samples. KNN full distances are computed as mN , whereas BP-KNN full distances are computed as $m|\Omega_i|$. ^a For MWS, operation counts are reported on the fixed stratified diagnostic subset used for permutation-sensitivity analysis.

4.3. Comparative Performance Results for the Case Study K-Means

As indicated by the results in Table 13, the incorporation of the BMS framework consistently enhanced or maintained the clustering accuracy across all evaluated distance metrics. In the Wine dataset, BP-K-means achieved a notable improvement, attaining 70.22% accuracy compared to the standard K-means' 65.73%. Similarly, in the Breast Cancer dataset, BP-K-means achieved 90.51% accuracy, surpassing the standard model's 85.41%. Unlike the standard algorithm, which demonstrated its lowest performance (65.73%) specifically with the Euclidean metric, BP-K-means exhibited remarkable stability, producing consistent results regardless of whether the Euclidean, Manhattan or Minkowski metrics were utilized. The primary advantage of BP-K-means lies in its capacity to reach optimal solutions in fewer iterations and with a reduced hardware overhead. In the CDC Diabetes dataset, while both models achieved an accuracy of 86.06%, BP-K-means converged within five to seven iterations, whereas the standard model required eight iterations. This acceleration was even more pronounced in the MWS dataset, where BP-K-means converged in 9–11 iterations, significantly outperforming the 16–17 iterations required by the standard model to achieve the same accuracy.

Table 13. Performance comparisons of BP-K-means and standard K-means on different datasets.

Model	Dataset	Distance Metric	Accuracy	F1-Macro	F1-Weighted	ROC
K-means	Wine	Euclidean	0.6573	0.5226	0.5662	0.7042
		Manhattan	0.7022	0.7031	0.7147	0.7747
		Chebyshev	0.7022	0.7031	0.7147	0.7747
		Minkowski	0.6573	0.5226	0.5662	0.7042
BP-K-means	Wine	Euclidean	0.7022	0.7031	0.7147	0.7747
		Manhattan	0.7022	0.7031	0.7147	0.7747
		Chebyshev	0.7022	0.7031	0.7147	0.7747
		Minkowski	0.7022	0.7031	0.7147	0.7747
K-means	Breast Cancer	Euclidean	0.8541	0.8268	0.8443	0.8052
		Manhattan	0.8506	0.8220	0.8402	0.8004
		Chebyshev	0.8506	0.8220	0.8402	0.8004
		Minkowski	0.8541	0.8268	0.8443	0.8052
BP-K-means	Breast Cancer	Euclidean	0.9051	0.8966	0.9042	0.8761
		Manhattan	0.9016	0.8885	0.8982	0.9247
		Chebyshev	0.8506	0.8966	0.9042	0.8004
		Minkowski	0.9051	0.8966	0.9042	0.8761
K-means	Iris	Euclidean	0.8926	0.8917	0.8910	0.9834
		Manhattan	0.8859	0.8847	0.8839	0.9827
		Chebyshev	0.8859	0.8857	0.8849	0.9810
		Minkowski	0.8926	0.8917	0.8910	0.9834
BP-K-means	Iris	Euclidean	0.8926	0.8917	0.8910	0.9834
		Manhattan	0.8859	0.8847	0.8839	0.9827
		Chebyshev	0.8859	0.8857	0.8849	0.9810
		Minkowski	0.8926	0.8917	0.8910	0.9834
K-means	CDC Diabetes	Euclidean	0.8606	0.4625	0.8606	0.5
		Manhattan	0.8606	0.4625	0.8606	0.5
		Chebyshev	0.8606	0.4625	0.8606	0.5
		Minkowski	0.8606	0.4625	0.8606	0.5
BP-K-means	CDC Diabetes	Euclidean	0.8606	0.4625	0.8606	0.5
		Manhattan	0.8606	0.4625	0.8606	0.5
		Chebyshev	0.8606	0.4625	0.8606	0.5
		Minkowski	0.8606	0.4625	0.8606	0.5

Table 13. Cont.

Model	Dataset	Distance Metric	Accuracy	F1-Macro	F1-Weighted	ROC
K-means	Abalone	Euclidean	0.2722	0.0855	0.2315	0.5372
		Manhattan	0.2760	0.0872	0.2365	0.5376
		Chebyshev	0.2724	0.0865	0.2408	0.5316
		Minkowski	0.2722	0.0855	0.2315	0.5372
BP-K-means	Abalone	Euclidean	0.2724	0.0820	0.2354	0.5304
		Manhattan	0.2726	0.0789	0.2327	0.5290
		Chebyshev	0.2712	0.0864	0.2392	0.5321
		Minkowski	0.2724	0.0855	0.2315	0.5304
K-means	MWS	Euclidean	0.6375	0.3893	0.4964	0.5000
		Manhattan	0.6375	0.3893	0.4964	0.5000
		Chebyshev	0.6375	0.3893	0.4964	0.5000
		Minkowski	0.6375	0.3893	0.4964	0.5000
BP-K-means	MWS	Euclidean	0.6375	0.3893	0.4964	0.5000
		Manhattan	0.6375	0.3893	0.4964	0.5000
		Chebyshev	0.6375	0.3893	0.4964	0.5000
		Minkowski	0.6375	0.3893	0.4964	0.5000

As demonstrated in Table 14, on the large-scale MWS dataset, BP-K-means maintained a constant RAM consumption (2713 MB) and reduced the processing time to as low as 6.38 s (Chebyshev). This efficiency confirms that the (B3) constraint prevents the center points from oscillating near unstable class boundaries, ensuring a more direct trajectory toward the global optimum.

The enhanced balance and efficiency of BP-K-means are fundamentally attributed to the bipolar transformation delineated in Equation (66).

$$x'_i = \text{sign}(x_i - x) \quad (66)$$

This transformation effectively recalibrates the distance relationships by compressing the data space into the $[-1, +1]$ interval. By distinctly differentiating between the positive and negative axes, the neighborhood and similarity measures ($d(x_i, x_j)$) are directly stabilized. As demonstrated in Tables 8, 9, 13 and 14, this recalibration enhances class balance, particularly in imbalanced datasets such as CDC Diabetes, while ensuring the preservation of the manifold's geometric integrity. Consequently, the BP-K-means model emerges as a highly applicable approach for large-scale datasets, optimizing the balance between time efficiency and clustering performance through accelerated convergence without compromising the underlying clustering efficacy.

The experimental results confirm the theoretical findings. The (B3) constraint acts as an implicit regularizer that filters out geometrically inconsistent samples. This leads to improved robustness, reduced variance, and better generalization. Moreover, the computational gains demonstrate that the proposed method is not only more accurate but also more efficient, making it suitable for large-scale applications.

To assess the significance of the observed performance improvements, a non-parametric Wilcoxon signed-rank test was employed at a confidence level of $\alpha = 0.05$. As shown in the Table 15, the comparison between F1-macro and ROC scores BP-KNN and standard KNN resulted in a p -value of 0.031. This finding statistically substantiates that the structural regularization afforded by the BMS consistently and significantly enhances classification accuracy across various datasets. Also, the Wilcoxon signed-rank test comparing BP-K-means with standard K-means yielded a p -value of 0.500. This result suggests that the two algorithms exhibit statistically equivalent clustering performance in terms of accuracy, F1-macro, and ROC scores. However, the proposed BP-K-means model

offers a significant advantage in computational efficiency. As evidenced in Tables 13 and 14, BP-K-means achieves this competitive equivalence with substantially fewer iterations and reduced hardware overhead, thereby establishing its efficacy as a high-speed alternative for large-scale applications.

Table 14. RAM and time consumptions of BP-K-means and standard K-means on different datasets.

Model	Dataset	Distance Metric	RAM Consumption (MB)	Time Consumption (s)
K-means	Wine	Euclidean	186.18	0.0011
		Manhattan	186.12	0.0025
		Chebyshev	186.08	0.0011
		Minkowski	186.18	0.0011
BP-K-means	Wine	Euclidean	185.61	0.0010
		Manhattan	185.35	0.0021
		Chebyshev	185.80	0.0010
		Minkowski	185.61	0.0010
K-means	Breast Cancer	Euclidean	198.40	0.017
		Manhattan	188.56	0.0020
		Chebyshev	188.68	0.0025
		Minkowski	198.40	0.017
BP-K-means	Breast Cancer	Euclidean	188.54	0.0132
		Manhattan	188.42	0.0098
		Chebyshev	188.31	0.0021
		Minkowski	188.54	0.0132
K-means	Iris	Euclidean	187.37	0.0092
		Manhattan	187.54	0.0022
		Chebyshev	187.93	0.0025
		Minkowski	187.37	0.0092
BP-K-means	Iris	Euclidean	187.20	0.0074
		Manhattan	187.50	0.0010
		Chebyshev	187.82	0.0019
		Minkowski	187.20	0.0074
K-means	CDC Diabetes	Euclidean	497.26	0.2776
		Manhattan	463.28	0.2623
		Chebyshev	466.01	0.2803
		Minkowski	497.26	0.2776
BP-K-means	CDC Diabetes	Euclidean	375.39	0.2150
		Manhattan	386.71	0.2385
		Chebyshev	281.89	0.2491
		Minkowski	375.39	0.2150
K-means	Abalone	Euclidean	194.98	0.0709
		Manhattan	194.76	0.0777
		Chebyshev	194.87	0.0790
		Minkowski	194.98	0.0709
BP-K-means	Abalone	Euclidean	193.66	0.0493
		Manhattan	193.55	0.0375
		Chebyshev	193.44	0.0314
		Minkowski	193.66	0.0493
K-means	MWS	Euclidean	2721.31	52.8058
		Manhattan	2767.35	33.1418
		Chebyshev	2824.86	18.4402
		Minkowski	2721.31	52.8058
BP-K-means	MWS	Euclidean	2713.28	15.9605
		Manhattan	2713.30	12.9524
		Chebyshev	2706.61	6.3761
		Minkowski	2713.28	15.9605

Table 15. Wilcoxon’s signed rank test results on case studies.

Case Study	<i>p</i> -Value
BP-KNN	0.031
BP-K-means	0.5

5. Discussion

The basic principle of the standard KNN algorithm is to compute the numerical distance between each query sample and all n training samples. For a d -dimensional feature vector, this requires a full numerical distance evaluation over all feature dimensions. After these distances are obtained, the k -nearest samples are selected for majority voting. Therefore, the computational structure of standard KNN can be written as

$$T_{\text{KNN}} = O(n \cdot d) + O(n \log k).$$

In the BP-KNN procedure, the algorithm does not perform a pairwise $O(k^2)$ relational check inside a preselected k -neighborhood. Instead, BP-KNN applies a bipolar sign-space structural filter over the candidate set and constructs the retained candidate set Ω_i . The final full numerical distance evaluation and ranking are then performed only among the retained candidates $a \in \Omega_i$. Therefore, the operational cost of BP-KNN is better represented as

$$T_{\text{BP-KNN}} = O(n \cdot d_{\text{sign}}) + O(|\Omega_i| \cdot d) + O(|\Omega_i| \log k),$$

where d_{sign} denotes the dimensionality of the bipolar sign-space representation and $|\Omega_i|$ denotes the number of retained candidates after the structural filter.

This formulation clarifies that BP-KNN should not be interpreted as reducing the worst-case $O(mn)$ candidate-scanning structure. In the worst case, the algorithm still scans the candidate set for each query instance. However, when the sign-space structural filter excludes a sufficient proportion of candidates, the number of final full numerical distance evaluations is reduced from n to $|\Omega_i|$, which leads to average-case and implementation-level efficiency gains.

This interpretation is conceptually related to inequality-based acceleration strategies such as OTI/EOTI [7], where geometric inequalities are used to reduce unnecessary distance evaluations. However, BP-KNN differs from OTI/EOTI in that it does not claim a sublinear nearest-neighbor search guarantee. Instead, BP-KNN uses a BMS-inspired bipolar sign-space structural filter to reduce the number of final full numerical distance evaluations before the final neighbor ranking stage.

Figure 3 provides a detailed comparison of the execution times and corresponding speedup factors across all six datasets. A logarithmic scale was employed to accommodate the multi-magnitude variance in the time scales. The annotations indicate the absolute execution time (in seconds) and relative speedup achieved by BP-KNN for both models. The practical computational advantage of the BMS framework becomes more pronounced as dataset size and dimensionality increase. In high-dimensional and large-volume scenarios, such as MWS and CDC Diabetes, BP-KNN achieved empirical speedups of $11.6\times$ and $4.3\times$, respectively. Therefore, the observed speedups are primarily attributed to candidate reduction before final numerical ranking, vectorized sign-space validation, and reduced memory allocation, rather than to a reduction in the worst-case $O(mn)$ candidate-scanning structure. This is consistent with the FLOP analysis, where the final full-distance cost is expressed as $N(1 - P_i) \cdot 3d$ after the structural filter.

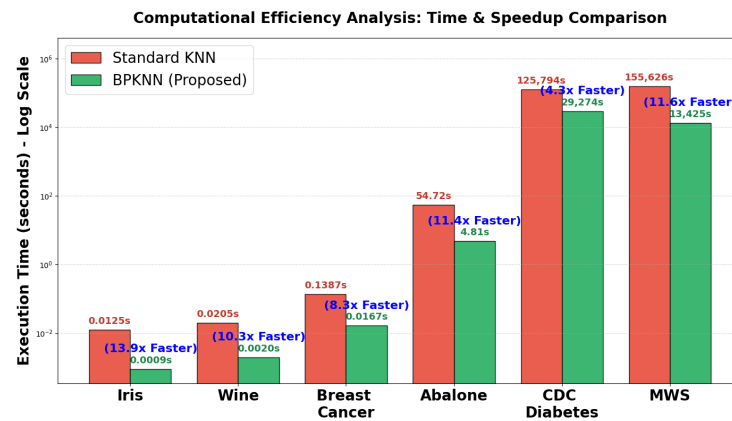


Figure 3. Computational efficiency of BP-KNN and standard KNN.

Figure 4 illustrates the memory scalability of our proposed framework. On smaller datasets (such as Iris and Wine), both algorithms demonstrated similar memory consumption (approximately 200 MB). However, as the dataset size (MWS) increased, the memory overhead of the standard KNN increased to 2713 MB, whereas BP-KNN achieved a highly efficient memory consumption of only 689 MB (approximately a 75% reduction). This result empirically supports the theory that the BP-KNN does not require full pairwise distance-matrix storage or processing capabilities.

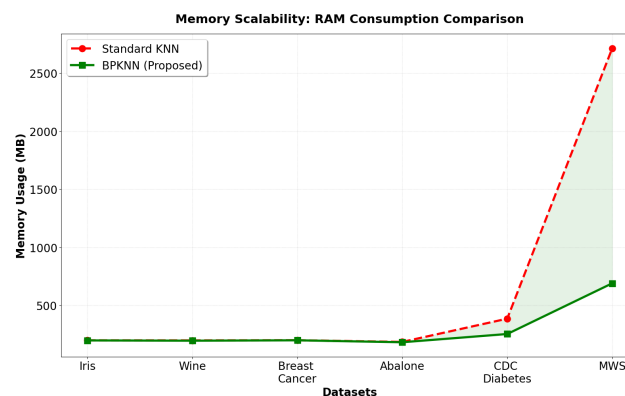


Figure 4. Memory scalability of BP-KNN and standard KNN.

Figure 5 offers a comparative analysis of decision stability under extreme Gaussian noise ($\sigma \in [0, 2.0]$) for the Breast Cancer dataset, affirming the robustness of the proposed BMS framework. The BP-KNN ($k = 5$) demonstrated decision invariance, consistently achieving an accuracy rate of 92.1% regardless of noise density. This horizontal trajectory empirically confirms that the (B3) quadrilateral inequality serves as a structural shield, effectively isolating the voting process from corrupted samples that violate geometric consistency. In contrast, the standard KNN ($k = 5$) displays stochastic instability. The accuracy of the underlying model remained constant at a lower threshold (91.2%) and exhibited unpredictable fluctuations, such as the transient jump at $\sigma = 1.6$. This jump underscores that the performance of the standard KNN is dictated by random spatial alignment rather than structural integrity. While the standard KNN is susceptible to feature degradation, BP-KNN filters high-frequency noise through its internal metric mechanism, ensuring that the decision boundary remains topologically bound to the actual data manifold. This robustness is crucial for reliable performance in real-world scenarios, where data sensors and acquisition processes are inherently exposed to varying noise levels. This explains why BP-KNN not only provides significantly faster inference and higher accuracy in practice but also provides more stable decision boundaries. The computational complexity of the

standard K-means algorithm is determined by two iterative steps, assignment and centroid update, over iteration t . The main cost is in the assignment step, where each of the n samples is compared with k cluster centroids in a d -dimensional space. Therefore, the total complexity is given as Equation (67):

$$T_{Kmeans} = \mathcal{O}(t_{standard} \cdot n \cdot k \cdot d) \quad (67)$$

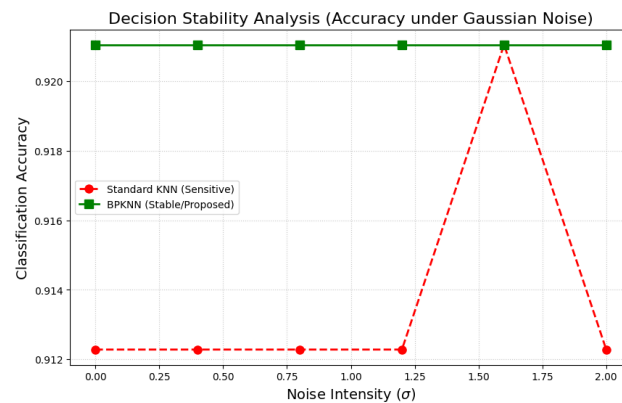


Figure 5. Decision stability of BP-KNN and standard KNN.

The BP-K-means algorithm fundamentally changes the assignment step by introducing the (B3) condition as a strict constraint. Unlike the soft-penalty approach, it completely prohibits assignments that violate geometric consistency. In each assignment step, the (B3) condition was checked for every point-centroid pair. This check imposes a small and constant time overhead. If the condition is violated, the cost of this assignment is assumed to be infinite, making it an infeasible assignment. Therefore, the per-iteration cost for BP-K-means remains asymptotically the same as that for standard K-means, as shown in Equation (68), because the additional checks do not change the dominant term:

$$C_{iterBP-K-means} = \mathcal{O}(n \cdot k \cdot d) \quad (68)$$

A critical advantage of this approach is its effect on the number of iterations required for convergence (t). The hard filtering mechanism prevents geometrically inconsistent assignments and avoids the reassignment of points at the cluster boundaries. This stabilization allows for a more direct and rapid convergence. Consequently, there was a significant reduction in the total number of iterations (t) required to obtain a stable solution. The total effective complexity of BP-K-means is presented in Equation (69).

$$T_{BP-K-means} = \mathcal{O}(t_{bipolar} \cdot n \cdot k \cdot d), \text{ where } (t_{bipolar} < t_{standard}) \quad (69)$$

The theoretical advantage of achieving convergence in fewer iterations is directly reflected in our experimental findings. According to the results of BP-K-means and standard K-means, standard K-means has an average processing time of 664 MB of RAM and 6.612 s, whereas BP-K-means has an average processing time of 630 MB of RAM and 2.183 s. This ratio demonstrates that BP-K-means is more efficient than the standard version. The reduction in RAM usage can be explained by the earlier convergence and, consequently, fewer core update histories stored in the memory. Therefore, theory and practice are fully compatible with each other. The (B3) condition introduces a small, intelligent overhead that provides significant gains by reducing the total number of iterations, making BP-K-means a more efficient overall model.

The theoretical and practical performances of the standard and proposed bipolar models are summarized in Table 16. As the analysis shows, BP-KNN does not reduce

the worst-case candidate-scanning structure; it still evaluates the candidate set through the sign-space structural filter. However, its computational advantage arises from reducing the number of final full numerical distance evaluations from n to $|\Omega_i|$ through the retained candidate set. Therefore, the gain should be interpreted as an average-case and implementation-level reduction in the final numerical ranking workload rather than a reduction in worst-case asymptotic complexity. For BP-K-means, no retained KNN candidate set is formed. t_{bipolar} denotes the number of Lloyd iterations after applying the hard ($B3$) assignment constraint.

Table 16. Complexity and RAM consumption summary table of standard and bipolar models.

Model	Additional Component	Complexity	RAM Consumption (Avg.)	Distance Calc. Cost
KNN	-	$\mathcal{O}(n \cdot d) + \mathcal{O}(n \log k)$	646 MB	$\approx 3d$ FLOPs
K-means	-	$\mathcal{O}(t \cdot n \cdot K \cdot d)$	664 MB	
BP-KNN	Bipolar sign-space structural filter	$\mathcal{O}(n \cdot d_{\text{sign}}) + \mathcal{O}(\Omega_i \cdot d) + \mathcal{O}(\Omega_i \log k)$	285 MB	$\approx 4 + (1 - P_i) \cdot 3d$ FLOPs.
BP-K-means	Hard ($B3$) assignment constraint	$\mathcal{O}(t_{\text{bipolar}} \cdot n \cdot K \cdot d)$	630 MB	Comparable per-iteration cost; fewer iterations

Here, d_{sign} denotes the dimensionality of the bipolar sign-space representation, and $|\Omega_i|$ denotes the retained candidate-set size after the structural filter.

Figure 6 presents a comparative efficiency analysis of standard KNN and BP-KNN on the Breast Cancer dataset with $k = 5$. Standard KNN has a 0% candidate-exclusion rate because it evaluates all training samples exhaustively. In contrast, BP-KNN applies the bipolar sign-space structural filter before the final numerical ranking. Under this filter, BP-KNN achieved an average candidate-exclusion rate of 18.26%, with local fluctuations ranging from approximately 12% to 23%. These fluctuations indicate that the proposed filter adapts to local structural heterogeneity by excluding more candidates in geometrically inconsistent regions while remaining conservative in stable regions. The shaded area in the graph represents the candidates excluded before the final numerical distance ranking.

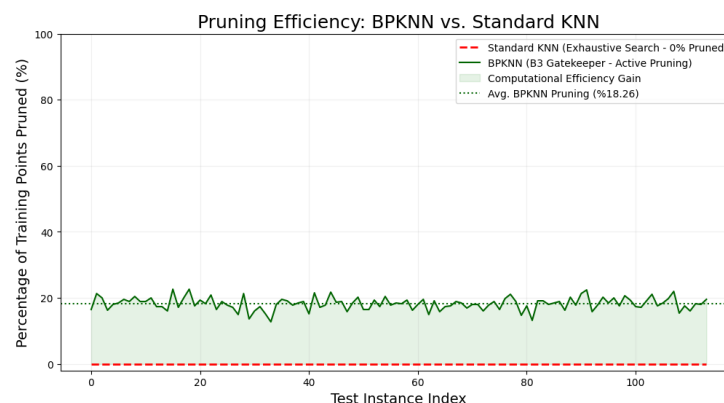


Figure 6. Pruning efficiency of BP-KNN and Standard KNN.

It is important to distinguish the contribution of candidate reduction from the larger wall-clock speedups observed in the experiments. The candidate-exclusion rates reported in Table 12 are approximately 16–20%. If all other implementation factors were identical, this reduction alone would correspond to a final full-distance workload reduction of

roughly $1/(1 - P)$, i.e., approximately $1.19\times$ – $1.25\times$ for $P \in [0.16, 0.20]$. Therefore, the candidate-reduction mechanism by itself should not be interpreted as fully explaining the larger empirical speedups reported in Figure 3, such as the $4.3\times$ and $11.6\times$ accelerations observed on CDC Diabetes and MWS. Rather, the wall-clock gains reflect the combined effect of several factors: fewer final full numerical distance evaluations, vectorized sign-space structural validation, reduced allocation of large dense distance arrays, improved cache locality, and differences in how the baseline exhaustive KNN distance ranking is materialized in memory. Thus, Table 12 quantifies the algorithmic candidate-reduction component, whereas the larger wall-clock speedups as an implementation-level acceleration are enabled by the two-stage BP-KNN computation graph.

To explain the empirical speedups observed in datasets such as Abalone and MWS despite the worst-case asymptotic overhead of the bipolar validation step, we further analyze the arithmetic-level computational cost in terms of Floating Point Operations (FLOPs). In standard distance-based learning, the Euclidean or Manhattan distance between two d -dimensional vectors requires operations over all feature dimensions. Ignoring constant implementation-level factors, a single full numerical distance evaluation requires approximately $3d$ elementary arithmetic operations per candidate pair. Therefore, for a query point evaluated against N training instances, standard KNN requires approximately

$$\text{FLOPs}_{\text{KNN}} \approx N \cdot 3d. \quad (70)$$

In the BP-KNN procedure, the computational graph is separated into two stages: a bipolar sign-space structural filter and a final numerical distance ranking. The sign-space filter first evaluates the indirect bipolar structural cost T_a^{sign} and determines whether each candidate satisfies

$$T_a^{\text{sign}} \leq \tau_i. \quad (71)$$

After the sign-space structural costs have been computed, cached, or structurally indexed, the runtime validation of the filter requires only retrieving the candidate structural cost, retrieving the query-specific threshold, performing one comparison, and updating the retained candidate set Ω_i . Thus, the four-FLOP interpretation refers specifically to this structural filter validation step, not to the full computation of Euclidean or Manhattan distances in the original continuous-valued feature space.

Let $P_i \in [0, 1]$ denote the candidate exclusion rate for query q_i :

$$P_i = \frac{N - |\Omega_i|}{N}. \quad (72)$$

Then, the number of final full numerical distance evaluations is reduced from N in standard KNN to $|\Omega_i| = N(1 - P_i)$ in BP-KNN. Accordingly, the approximate FLOP count for BP-KNN can be expressed as

$$\text{FLOPs}_{\text{BP-KNN}} \approx N \cdot 4 + N(1 - P_i) \cdot 3d. \quad (73)$$

This formulation clarifies why the bipolar filter becomes increasingly advantageous in higher-dimensional datasets. For example, in the MWS dataset with $d = 27$, a full numerical distance evaluation requires approximately $3d \approx 81$ arithmetic operations per candidate, whereas the cached structural filter validation requires approximately 4 operations. Therefore, when P_i is sufficiently large, the term $N(1 - P_i) \cdot 3d$ is substantially reduced, explaining the observed runtime and memory-efficiency gains without claiming a reduction in worst-case asymptotic complexity.

Averaging over all query samples gives the empirical pruning rate

$$P = \frac{1}{m} \sum_{i=1}^m P_i.$$

Thus, P represents the average proportion of candidates excluded by the bipolar sign-space structural filter before the final numerical distance ranking. This directly links the pruning-rate analysis in Figure 6 to the FLOP formulation in Equation (73).

In the case of K-means, the standard computation over t_{standard} Lloyd iterations scales as

$$T_{\text{Kmeans}} = O(t_{\text{standard}} \cdot n \cdot K \cdot d), \quad (74)$$

where n denotes the number of samples, K denotes the number of clusters, and d denotes the feature dimensionality. In BP-K-means, the BMS-inspired (B3) condition is incorporated into the assignment step as a hard geometric feasibility constraint for point-centroid assignments. Unlike BP-KNN, however, BP-K-means does not form a retained nearest-neighbor candidate set before a final numerical ranking stage. Therefore, the pruning-rate term P used to describe the retained-candidate mechanism of BP-KNN should not be directly interpreted as an identical candidate-pruning mechanism for BP-K-means.

Instead, the computational benefit of BP-K-means is mainly explained by assignment stabilization. The (B3)-based feasibility check prevents geometrically inconsistent point-centroid assignments and reduces boundary oscillations during centroid updates. Consequently, the algorithm may converge in fewer Lloyd iterations than standard K-means, i.e.,

$$t_{\text{BP-Kmeans}} < t_{\text{standard}}. \quad (75)$$

Accordingly, the effective workload of BP-K-means can be expressed as

$$T_{\text{BP-Kmeans}} = O(t_{\text{BP-Kmeans}} \cdot n \cdot K \cdot d), \quad t_{\text{BP-Kmeans}} < t_{\text{standard}}. \quad (76)$$

This interpretation indicates that the shorter processing times observed for BP-K-means are primarily associated with fewer assignment-update iterations and more stable centroid trajectories, rather than with the same retained-candidate pruning mechanism used in BP-KNN. Thus, BP-KNN and BP-K-means benefit from the same BMS-based geometric principle, but their computational savings arise through different operational mechanisms: BP-KNN reduces the number of final full-distance evaluations by constructing a retained candidate set, whereas BP-K-means reduces the global assignment-update workload by improving convergence stability.

In conclusion, both bipolar models exhibit more stable and efficient computations in practice. This improvement is achieved not by changing the worst-case asymptotic order, but by using the (B3) condition to suppress structurally inconsistent neighbor candidates in BP-KNN and geometrically infeasible point-centroid assignments in BP-K-means.

Limitations and Failure Cases

While the BP-KNN and BP-K-means algorithms offer significant advantages in complex, noisy, and unstable environments, they also encounter instances of failure. A critical evaluation of the framework uncovers several failure cases and theoretical limitations that require attention. Firstly, a fundamental limitation concerning computational efficiency emerges in perfectly linearly separable, zero-noise, and low-dimensional datasets like Iris. In such ideal scenarios, classical distance measures naturally establish optimal decision boundaries without needing additional structural adjustments. Consequently, the (B3) quadrilateral inequality validation becomes computationally redundant. The additional

sign-space structural filter validation in BP-KNN or the constant-time overhead per assignment in BP-K-means may outweigh the benefits of candidate-set reduction in BP-KNN or iteration reduction in BP-K-means. In these cases, bipolar models may exhibit slightly longer execution times than highly optimized standard classical algorithms without providing any additional marginal gain in accuracy. Secondly, the current BMS framework is designed for distance metrics based on absolute geometric quantities, particularly the Minkowski family (Euclidean, Manhattan, Chebyshev). Scale-independent or angular-based metrics, such as correlation distance and cosine similarity, were deliberately excluded from this study. Since correlation distance operates on angular alignments rather than spatial quantities, directly applying the quadrilateral condition ($B3$) often fails to produce a geometrically consistent bipolar space. Adapting bipolar axioms to correlation-based or non-Euclidean distances requires a hypersphere-specific polar mapping. This limits the applicability of the currently proposed algorithms in areas where angular proximity is preferred (e.g., certain text mining applications).

The third significant limitation, as clearly demonstrated in the experimental results (Table 13), is the identical prediction performance observed between BP-K-means and standard K-means on certain datasets, such as Iris, CDC Diabetes, and MWS. In these instances, both algorithms produced exactly the same classification metrics (accuracy, F1-Macro, and ROC). This empirical equivalence arises directly from the mathematical principles of the k-means working mechanism and the bipolar-filtering process. Although the quadrilateral inequality ($B3$) serves as a topological regulator, it is rarely violated if the dataset's natural spatial distribution is inherently free from significant geometric violations or if the initial values naturally fall into a stable region. Consequently, the constrained bipolar viable set becomes identical to the unconstrained set ($\Omega_{\text{Bipolar}} \equiv \Omega_{\text{Standard}}$). In such scenarios, BP-K-means reduces to the same level as standard K-means, and the bipolar structure does not diminish the algorithm's underlying performance. However, it cannot magically introduce new discriminative information into the feature space to enhance accuracy. Furthermore, this performance underscores a fundamental dependency limitation. The bipolar framework acts as a geometric filter rather than a feature extractor. In highly complex or heavily overlapping multi-class datasets (e.g., Abalone, where standard Manhattan K-means achieves 27.60% compared to BP-K-means' 27.26%), applying the strict ($B3$) constraint can sometimes over-regulate the boundaries if the underlying baseline metric fundamentally fails to separate the classes. This over-regulation can lead to negligible but existent accuracy drops or force the model to inherit the same poor clustering decisions (as seen in CDC Diabetes with an F1 score of 0.4625 for both models). This confirms that the capacity of BMS on K-means to accelerate convergence and ensure structural stability while improving absolute accuracy is fundamentally limited by the discriminative power of the selected baseline distance metric.

The empirical evaluation in this study follows a tiered design. Iris, Wine, and Breast Cancer are used as controlled low-dimensional benchmark datasets to examine the baseline behavior of the proposed bipolar structural filter under reproducible conditions. However, the evaluation is not limited to these small benchmarks. Abalone, CDC Diabetes, and MWS are included to assess the proposed framework under mixed-type, imbalanced, heterogeneous, and large-scale structured-data conditions. In particular, CDC Diabetes and MWS provide substantially larger and more realistic evaluation settings than the small benchmark datasets.

We acknowledge that applying the BMS-based structural filter to dense semantic embeddings extracted from pretrained deep networks is an important future direction. Such an evaluation would require a separate computer-vision-oriented experimental pipeline, including backbone selection, image normalization, embedding extraction, embedding-

space preprocessing, and a dedicated validation protocol. Therefore, CIFAR-100 or similar deep visual embedding benchmarks are beyond the scope of the present manuscript, whose primary objective is to introduce and analyze BMS as a geometric regularization operator for classical distance-based learning. Future work will investigate BP-KNN and related bipolar constraints as backbone-independent geometric filters operating on pretrained deep embeddings. Also, future work will investigate order-invariant or batch-aware anchor-selection strategies, because the current deterministic cyclic rule may become less reliable when query ordering reflects heterogeneous batches, class-imbalanced blocks, or distribution-shifted acquisition regimes.

A full reproducible benchmark against exact implementations of OTI/EOTI, MCML/BMCML, and NNS-DML under a unified tuning, preprocessing, and hardware protocol remains an important future extension.

6. Conclusions

This study comprehensively explored the potential of an innovative methodology based on the BMS concept to enhance the performance, generalizability, and efficiency of the standard KNN and K-means models. The underlying principle of the quadrilateral inequality (the $(B3)$ condition) provides geometric integrity to classical distance-based algorithms, thereby strengthening the structural consistency of their decision-making mechanisms. Integrating this principle at the algorithmic level demonstrates that the method possesses transformative potential, although its effectiveness naturally varies depending on the data type and problem structure.

Experimental analyses conducted on six diverse datasets demonstrate that the BP-KNN provides significant predictive and efficiency gains, whereas BP-K-means mainly provides computational efficiency and convergence-stability advantages while maintaining competitive clustering performance. For instance, the BP-KNN's approximately 11-fold faster inference time on the multi-class and complex Abalone dataset clearly illustrates that the bipolar approach improves not only classification accuracy but also computational efficiency. Similarly, the BP-K-means model produced more balanced clusters on the Wine dataset, yielding a substantial improvement in the macro F1-score.

These findings indicate that the BMS framework delivers more than just a metric generalization over standard KNN and K-means approaches; it introduces a fundamental structural enhancement. In this context, the bipolar approach provides robust solutions for critical challenges such as data imbalance, geometric regularization, structural consistency, and optimization stability. Consequently, this study highlights the crucial importance of integrating geometric integrity as a core component of the learning process within the broader machine learning literature.

Author Contributions: Conceptualization, G.M.; methodology, G.M.; software, G.M.; validation, G.M., Ç.İ.A.; formal analysis, G.M., Ç.İ.A.; investigation, G.M.; resources, G.M., Ç.İ.A.; data curation, Ç.İ.A.; writing—original draft preparation, G.M.; writing—review and editing, G.M., Ç.İ.A.; visualization, G.M., Ç.İ.A.; supervision, Ç.İ.A.; project administration, Ç.İ.A.; funding acquisition, Ç.İ.A. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the Scientific and Technological Research Council of Türkiye (TUBITAK) within the 1002 Support Program (Grant No: 124E297). The APC was funded by the authors.

Data Availability Statement: The implementation source code required to reproduce the experiments reported in this study is publicly available at (accessed on 13 July 2026) <https://github.com/gizenmtl/bipolar-metric-learning-reproducibility>. The repository includes the manuscript-specific BP-KNN and BP-K-means implementations, baseline comparison scripts, metric-specific experimental routines,

operation-count and candidate-reduction analysis scripts, permutation-sensitivity analysis scripts, preprocessing steps, environment requirements, and reproduction instructions. The raw datasets are not redistributed because they are publicly available from their original sources.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

BMS	Bipolar Metric Spaces
BP-K-Means	Bipolar K-Means
BP-KNN	Bipolar K-Nearest Neighbor
FLOPs	Floating Point Operations
KNN	K-Nearest Neighbor
MWS	Malicious Website Features

References

1. Suárez, J.L.; García, S.; Herrera, F. A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges. *Neurocomputing* **2021**, *425*, 300–322. <https://doi.org/10.1016/j.neucom.2020.08.017>.
2. Yuan, B.W.; Luo, X.G.; Zhang, Z.L.; Yu, Y.; Huo, H.W.; Johannes, T.; Zou, X.D. A novel density-based adaptive k nearest neighbor method for dealing with overlapping problem in imbalanced datasets. *Neural Comput. Appl.* **2021**, *33*, 4457–4481. <https://doi.org/10.1007/s00521-020-05256-0>.
3. Amer, A.A.; Ravana, S.D.; Habeeb, R.A.A. Effective k-nearest neighbor models for data classification enhancement. *J. Big Data* **2025**, *12*, 86. <https://doi.org/10.1186/s40537-025-01137-2>.
4. Gutiérrez, P.D.; Lastra, M.; Bacardit, J.; Benítez, J.M.; Herrera, F. GPU-SME-kNN: Scalable and memory efficient kNN and lazy learning using GPUs. *Inf. Sci.* **2016**, *373*, 165–182. <https://doi.org/10.1016/j.ins.2016.08.089>.
5. Gao, H.; Ma, Z. Geometric Metric Learning for Multi-Output Learning. *Mathematics* **2022**, *10*, 1632. <https://doi.org/10.3390/math10101632>.
6. Halder, R.K.; Uddin, M.N.; Uddin, M.A.; Aryal, S.; Khraisat, A. Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications. *J. Big Data* **2024**, *11*, 113. <https://doi.org/10.1186/s40537-024-00973-y>.
7. Pan, Y.; Pan, Z.; Wang, Y.; Wang, W. A new fast search algorithm for exact k-nearest neighbors based on optimal triangle-inequality-based check strategy. *Knowl.-Based Syst.* **2020**, *189*, 105088. <https://doi.org/10.1016/j.knosys.2019.105088>.
8. Yang, X.; Chen, H.; Li, T.; Wan, J.; Sang, B. Neighborhood rough sets with distance metric learning for feature selection. *Knowl.-Based Syst.* **2021**, *224*, 107076. <https://doi.org/10.1016/j.knosys.2021.107076>.
9. Ruan, Y.; Xiao, Y.; Hao, Z.; Liu, B. A nearest-neighbor search model for distance metric learning. *Inf. Sci.* **2021**, *552*, 261–277. <https://doi.org/10.1016/j.ins.2020.11.054>.
10. Ruan, Y.; Xiao, Y.; Hao, Z.; Liu, B. A convex multi-class model via distance metric learning based class-to-instance confidence. *Knowl.-Based Syst.* **2022**, *258*, 109791. <https://doi.org/10.1016/j.knosys.2022.109791>.
11. Zhang, X.; Xiao, H.; Gao, R.; Zhang, H.; Wang, Y. K-nearest neighbors rule combining prototype selection and local feature weighting for classification. *Knowl.-Based Syst.* **2022**, *243*, 108451. <https://doi.org/10.1016/j.knosys.2022.108451>.
12. Yuan, C.; Zhou, C.; Peng, J.; Li, H. Mixture correntropy-based robust distance metric learning for classification. *Knowl.-Based Syst.* **2024**, *295*, 111791. <https://doi.org/10.1016/j.knosys.2024.111791>.
13. Kulis, B. Metric Learning: A Survey. *Found. Trends® Mach. Learn.* **2013**, *5*, 287–364. <https://doi.org/10.1561/22000000019>.
14. Weinberger, K.Q.; Saul, L.K. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *J. Mach. Learn. Res.* **2009**, *10*, 207–244.
15. Celebi, M.E.; Kingravi, H.A.; Vela, P.A. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl.* **2013**, *40*, 200–210. <https://doi.org/10.1016/j.eswa.2012.07.021>.
16. Xu, R.; WunschII, D. Survey of Clustering Algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. <https://doi.org/10.1109/TNN.2005.845141>.
17. Huang, X.; Ye, Y.; Guo, H.; Cai, Y.; Zhang, H.; Li, Y. DSKmeans: A new kmeans-type approach to discriminative subspace clustering. *Knowl.-Based Syst.* **2014**, *70*, 293–300. <https://doi.org/10.1016/j.knosys.2014.07.009>.

18. Capó, M.; Pérez, A.; Lozano, J.A. An efficient approximation to the K-means clustering for massive data. *Knowl.-Based Syst.* **2017**, *117*, 56–69. <https://doi.org/10.1016/j.knosys.2016.06.031>.
19. Kim, H.; Kim, H.K.; Cho, S. Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling. *Expert Syst. Appl.* **2020**, *150*, 113288. <https://doi.org/10.1016/j.eswa.2020.113288>.
20. Zhu, A.; Hua, Z.; Shi, Y.; Tang, Y.; Miao, L. An Improved K-Means Algorithm Based on Evidence Distance. *Entropy* **2021**, *23*, 1550. <https://doi.org/10.3390/e23111550>.
21. Ay, M.; Özbakır, L.; Kulluk, S.; Gülmez, B.; Öztürk, G.; Özer, S. FC-Kmeans: Fixed-centered K-means algorithm. *Expert Syst. Appl.* **2023**, *211*, 118656. <https://doi.org/10.1016/j.eswa.2022.118656>.
22. Mutlu, A.; Gürdal, U. Bipolar metric spaces and some fixed point theorems. *J. Nonlinear Sci. Appl.* **2016**, *09*, 5362–5373. <https://doi.org/10.22436/jnsa.009.09.05>.
23. Dai, Q.; Liu, J.W.; Shi, Y.H. Distance mapping overlap complexity metric for class-imbalance problems. *Appl. Soft Comput.* **2024**, *163*, 111904. <https://doi.org/10.1016/j.asoc.2024.111904>.
24. Laskhmaiah, K.; Krishna, S.M.; Reddy, B.E. An Optimized K-means with Density and Distance-Based Clustering Algorithm for Multidimensional Spatial Databases. *Int. J. Comput. Netw. Inf. Secur.* **2021**, *13*, 70–82. <https://doi.org/10.5815/ijcnis.2021.06.06>.
25. Abdalnassar, A.; Nair, L.R. Performance analysis of Kmeans with modified initial centroid selection algorithms and developed Kmeans9+ model. *Meas. Sens.* **2023**, *25*, 100666. <https://doi.org/10.1016/j.measen.2023.100666>.
26. Xia, S.; Peng, D.; Meng, D.; Zhang, C.; Wang, G.; Glem, E.; Wei, W.; Chen, Z. Ball k-means: A Fast Adaptive k-means with No Bounds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 87–99. <https://doi.org/10.1109/TPAMI.2020.3008694>.
27. Rajput, P.P. Fuzzy K-Means Clustering with Discriminative Embedding: An Implementation. *Int. J. Res. Appl. Sci. Eng. Technol.* **2024**, *12*, 2092–2098. <https://doi.org/10.22214/ijraset.2024.63436>.
28. Xie, Z.; Nikolayeva, O.; Luo, J.; Li, D. Building Risk Prediction Models for Type 2 Diabetes Using Machine Learning Techniques. *Prev. Chronic Dis.* **2019**, *16*, 190109. <https://doi.org/10.5888/pcd16.190109>.
29. Warwick Nash, T.S. Abalone, 1994. Available online: <https://archive.ics.uci.edu/dataset/1/abalone> (accessed on 15 February 2025).
30. Siddhartha, M. Malicious Website Features, 2023. Available online: <https://huggingface.co/datasets/FredZhang7/malicious-website-features-2.4M> (accessed on 30 March 2025) .

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.