GBkNN-JGE: Enhancing GBkNN via Principle of Justifiable Granularity and Ensemble Learning

Manqi Lin
Wuhan University, Wuhan, China
nian0216lmq@gmail.com

Abstract. Achieving efficiency, robustness and interpretability in classification continues to pose significant challenges in data analysis. To address these issues, the granular-ball-based k-nearest neighbors (GBkNN) classifier has recently been proposed and demonstrated promising results. However, the performance of GBkNN heavily relies on the quality of GBs, and existing GB generation methods often use only purity as the evaluation criterion and apply a fixed threshold-based stopping rule. These limitations restrict their effectiveness in practical applications. To overcome this, we extend the advanced unsupervised GB generation method based on the Principle of Justifiable Granularity (POJG) to the supervised classification setting, aiming to enhance the overall performance of GBkNN. Furthermore, to mitigate the instability caused by the inherent randomness of k-means clustering, we propose a novel model named GBkNN-JGE under the ensemble learning paradigm. Experimental results on nine publicly available datasets show that our proposed method achieves superior performance compared to existing GB-based classifier and traditional machine learning methods.

Keywords: Granular-ball computing; Classification; Ensemble learning; k-Nearest neighbors.

1. Introduction

Classification, which aims to assign instances into predefined categories so that instances with the same label share common characteristics, is a fundamental supervised learning task in machine learning. In recent years, a variety of classification algorithms have emerged, such as decision trees [1], support vector machines (SVMs) [2], random forests [3], k-nearest neighbor [4], and neural networks [5]. Despite their effectiveness, these algorithms may encounter limitations in scalability and robustness when applied to increasingly large datasets. For example, k-nearest neighbor has the advantage that it is easy to implement. However, they are usually quite slow if the input data set is very large [6]. Neural networks have been widely applied as a flexible method in many fields. However, designing a suitable network for a specific task involves several key choices, such as the learning algorithm, network structure, number of layers, and neurons per layer. Consequently, how to perform classification efficiently and robustly, particularly on large-scale datasets, remains a challenging problem.

Recently, GBkNN has emerged as a promising classification method that effectively addresses challenges in efficiency, robustness, and interpretability [7]. GBkNN is built upon the framework of GB computing, which originates from the idea of granular computing—a paradigm that processes data using information granules rather than individual instances. In GB computing, a GB is a hypersphere that encloses a group of similar data instances, serving as a higher-level abstraction for computation. Compared to traditional instance-based methods, GBkNN reduces computational cost by performing classification on GBs instead of individual points, thereby improving scalability while preserving interpretability. Moreover, its multi-granularity structure allows the classifier to adapt flexibly to different data distributions, which enhances its robustness in real-world applications.

The performance of GBkNN relies heavily on the quality of the granular balls, which in turn depends on the GB generation method. Ideally, the generated GBs should align well with the underlying structure of the data [8]. However, most existing GB generation approaches in supervised learning suffer from several limitations [9]. Specifically, they often evaluate GB quality solely based on purity—that is, the proportion of majority-class instances within a GB—and use a fixed threshold (e.g., 0.99) as the stopping criterion for splitting. This single-indicator, threshold-based strategy tends to generate overly small GBs, which fails to capture meaningful structures in the data and may lead to poor generalization. Consequently, when the generated GBs do not reflect the actual distribution of data, classification results may become inaccurate. These limitations highlight the need for more comprehensive and stable GB generation strategies to fully exploit the potential of GB-based classifiers.

The main aim of this article is to present an effective GB-based model for classification tasks, aiming to solve the aforementioned problems. The main contributions of this article are as follows.

1)The advanced unsupervised GB-POJG method [10] is extended to the supervised learning domain by incorporating label information directly into the GB construction process. Instead of using a fixed purity threshold to terminate GB splitting, the proposed method evaluates GB quality in a more adaptive and comprehensive way, which allows better alignment with the data distribution.

2) To mitigate the instability caused by the randomness of k-means clustering in GB generation, an ensemble learning strategy is introduced. By constructing multiple classifiers and leveraging a weighted evaluation method, our method ensures more stable and reliable classification outcomes.

3) By integrating these procedures, a novel GB-based model, termed GBkNN-JGE, is proposed for classification tasks, effectively solving existing problems in GB based classification algorithms.

The remainder of this article is organized as follows. Section 2 presents a comprehensive and focused review of related works. Section 3 described GBkNN-JGE in detail. Section 4 covers a series of numerical experiments conducted on publicly available datasets to validate the performance of GBkNN-JGE. Finally, the conclusions and further work are presented in Section 5.

2. Related Works

To make the paper more concise, in the subsequent content, let $D=\{(\mathbf{x_1},y_1),(\mathbf{x_2},y_2),...,(\mathbf{x_n},y_n)\}$ be a dataset. Let $\mathbf{X}=[\mathbf{x_1},\mathbf{x_2},...,\mathbf{x_n}]\in\mathbb{R}^{m\times n}$ represents the feature matrix of D with m features and n instances, and $\mathbf{y}=[y_1,y_2,...,y_n]\in\mathbb{R}$) represents the corresponding label vector.

2.1 Granular Ball Computing

To begin with, several essential definitions concerning GB computing are reviewed as follows.

Definition 1(see [7]): Given a dataset *D*, let *U* be a subset of *D*. The GB derived from *U* with center ***c*** and radius *r*, denoted as Ω_U, is defined by

$$\mathbf{c}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{x_i};r=\frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x_i}-\mathbf{c}\right\| \tag{1}$$

where $\|\bullet\|$ is the 2-norm.

Definition 2(see [7]): Given a point ***o*** and a GB Ω_U with center ***c*** and radius r. The distance between ***o*** and Ω_U, denoted as dist(***o***, Ω_U), is defined as:

$$\mathrm{dist}(\mathbf{o},\Omega_U)=\begin{cases}\|\mathbf{o}-\mathbf{c}\|-r, \text{ if } \|\mathbf{o}-\mathbf{c}\|-r>0,\\ 0 \text{ , otherwise.}\end{cases} \tag{2}$$

Definition 3(see [7]): The purity of GB Ω_U is the proportion of majority-class instances in Ω_U, which is denoted as *p*.

2.2 Generating Granular Balls

The main concept behind the GB generation is to represent a dataset using a collection of GBs, each of which functions as an IG. The process in the GB generation method proposed by Xia [7] follows the steps illustrated in Fig. 1.

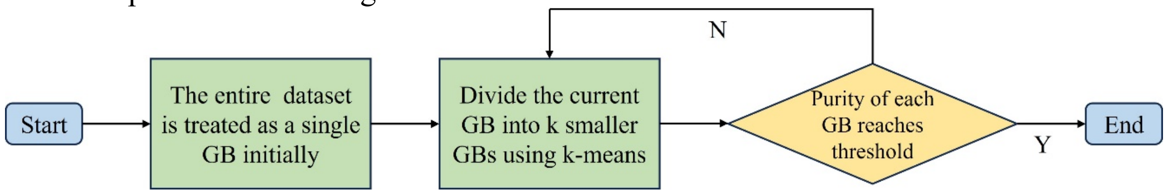

Fig. 1: Flowchart of the original GB generation algorithm

Evidently, the quality of GB is measured using the purity defined as Definition 3. The closer the purity is to 1, the more qualified the GB is. However, this method is discarded in this paper while a more comprehensive strategy is employed.

Grounded on the principle of justifiable granularity (POJG), Jia et al. proposed a new GB generation method named GB-POJG, which utilizes a comprehensive metric to assess GB quality and a strategy of maximizing overall quality to generate GB [10]. According to GB-POJG, the quality level of a GB is formally defined as follows.

Definition 4(see[10]): Let Ω_U be a GB with center $\boldsymbol{c}$ and radius r derived from U. The coverage of Ω_U is given by

$$Q_c(\Omega_U) = f_1(|\{\mathbf{x} \in \mathbf{X} : \|\mathbf{x} - \mathbf{c}\| \le r\}|) \tag{3}$$

where $f_1(t) = t$ and $|\cdot|$ denotes the cardinality of a set. The specificity of Ω_U is given by

$$Q_s(\Omega_U) = f_2(r) \tag{4}$$

where $f_2(t) = \exp(-\gamma t),\ \gamma \ge 0$. Further, the quality level of Ω_U is defined as

$$Q(\Omega_U) = Q_c(\Omega_U) \cdot Q_s(\Omega_U). \tag{5}$$

The GB Ω_U can be divided into Ω_{U_α} with center $\mathbf{c_\alpha}$ and Ω_{U_β} with center $\mathbf{c_\beta}$,where

$$U_\alpha = \{(\mathbf{x_i}, y_i) \in U : \|\mathbf{x_i} - \mathbf{c_\alpha}\| \le \|\mathbf{x_i} - \mathbf{c_\beta}\|\},\ U_\beta = U - U_\alpha \tag{6}$$

Typically, the GB generation algorithm follows a top-down greedy strategy to split GBs derived from the initial dataset. However, this method do not ensure the overall quality of generated GBs. In [10], a method of maximizing the overall quality of generated GBs was proposed. After generating a binary tree of GBs by a kind of 2-division method, a pruning strategy is applied to the binary tree to refine the generated GBs.

Definition 5(see[10]): Let Ω_U be a GB. If the number of instances belonging to Ω_U is sufficiently small, i.e., less than or equal to a threshold, then the best quality level of Ω_U , denoted by $BQ(\Omega_U)$, is defined as $BQ(\Omega_U) = Q(\Omega_U)$. Otherwise, if the number of instances is not small enough, the best quality level of Ω_U is defined as

$$BQ(\Omega_U) = \max(Q(\Omega_U), BQ(\Omega_{U_\alpha}) + BQ(\Omega_{U_\beta})) \tag{7}$$

where Ω_{U_α} and Ω_{U_β} represent the sub-GBs obtained by dividing Ω_U, and the threshold of the number of instances belonging to a GB is typically set as $\delta \cdot \sqrt{|U|}, \delta \in (0,1]$.

To identify the sub-GBs that lead to a GB’s best quality level, the definition of the best sub-GB combination is introduced.

Definition 6(see[10]): Let Ω_U be a GB. If $BQ(\Omega_U) = Q(\Omega_U)$, then the best combination of sub-GBs of Ω_U, denoted by $BC(\Omega_U)$, is defined as

$$BC(\Omega_U) = \{\Omega_U\} \tag{8}$$

Otherwise, $BC(\Omega_U)$ is defined as

$$BC(\Omega_U) = BC(\Omega_{U_\alpha}) \bigcup BC(\Omega_{U_\beta}) \tag{9}$$

2.3 Original Granular Ball k-nearest neighbors

The core idea of GBkNN is to assign a label to each queried instance based on the nearest GB, where the GB's label is determined by the majority vote of the instances it contains. The distance between a queried instance and a GB is defined according to Definition 2.

Thus, similar to traditional kNN, GBkNN classifies a queried instance based on nearby training information. However, unlike kNN, GBkNN does not need to select the parameter k.

3. The Proposed Method

This section first proposes a GB generation method that effectively leverages label information from classification tasks and employs a binary tree pruning strategy to maximize the overall quality of the generated GBs. Subsequently, the original GBkNN model is optimized by integrating ensemble learning techniques to further enhance performance. Finally, the complete GB generation algorithm and its time complexity are presented.

3.1 Generating Granular Balls with Maximum Overall Quality

As analyzed in Section 2.2, the 2-means clustering algorithm combined with GB-POJG is used to generate GBs. According to GB-POJG, two parameters, γ and δ, should be set in the model. However, since GB-POJG is mainly designed for clustering tasks involving unlabeled instances, this section introduces a new termination criterion for pre-division of GBs: when the purity of a GB is sufficiently high. As a result, the parameters that need to be set are changed to γ and p. Additionally, the termination criterion in GB-POJG is also used in this paper: when further division of GBs does not improve their overall quality.

Building on the above analysis, the task of determining the GBs generated from the dataset D reduces to finding the best combination of sub-GBs of Ω_D, which is derived from D. Since the GBs are divided using the 2-means clustering method, a binary tree structure is adopted to represent the relationships among GBs, enabling efficient organization and generation. Furthermore, a pruning strategy is employed on the binary tree to improve the quality of the resulting GBs. The detailed steps of this approach are described below.

Step 1: Initialize an empy tree T and assign GB Ω_D as the root node of T.

Step 2: For any node Ω_U in T, if the purity of GB is less than threshold p, divide Ω_U into Ω_{U_α} and Ω_{U_β} using 2-means clustering algorithm. Set Ω_U as the parent node of Ω_{U_α} and Ω_{U_β}.

Step 3: If the purity of GB of any leaf node in T is lower than or equal to threshold *p*, then continue the division process (Step 2). Otherwise, stop the process.

Step 4: Compute the quality of each node in T according to Eq. (3)–(5). Then, initialize the best sub-GB combination for each node based on Eq. (7).

Step 5: For any leaf node Ω_A in T, if its sibling node Ω_B is also a leaf node, then calculate the best quality level and best combination of sub-GBs of their parent node Ω_C using Eq. (6) and (8). After updating, prune both leaf nodes Ω_A and Ω_B from T.

Step 6: If more than one node remains in T, then repeat Step 5. Otherwise, terminate the process.

3.2 Enhancing GB*k*NN via Ensemble Learing

Although the GB*k*NN model demonstrates promising classification performance by leveraging GB structures, it still suffers from instability due to the random initialization inherent in the *k*-means clustering process used during GB generation. Specifically, different runs of *k*-means can produce significantly different granular structures even on the same dataset, leading to inconsistent classification outcomes. To address this issue, an ensemble learning framework is introduced to enhance the stability and performance of the GB*k*NN model.

In the proposed ensemble GB*k*NN framework, multiple base classifiers are constructed by repeatedly generating GBs using different random seeds during the *k*-means clustering stage. Each base classifier independently performs classification based on GBs it constructs. For each individual classifier, the label of an instance is determined by the nearest GB. Specifically, the instance inherits the label of the GB whose center is closest to it in the feature space. In the ensemble phase, instead of using the purity of a GB as in the traditional GB*k*NN model to asses its quality, a new method is employed according to Eq. (3)-(5). This measure simultaneously accounts for both the specificity and the coverage of a GB. For a given test instance, each base classifier searches for the nearest GB and outputs its best quality and label. Finally, the output of the ensemble classifier is defined as:

$$S(c)=\sum_{i=1}^{m}\mathrm{I}[y_i=c]\cdot BQ(\Omega_i);\quad \hat{y}=\arg\max_{c\in \mathrm{C}} S(c) \tag{10}$$

where m is the number of base classifiers, Ω_{U_i} and y_i is the output of i-th base classifier, $S(c)$ denotes the ensemble score for each possible class c, $\mathrm{C}=\{c_1,c_2,...,c_l\}$ denotes the set of all class labels. The class with the highest ensemble score is selected as the final output.

This ensemble approach effectively integrates the predictions of multiple diverse classifiers while emphasizing high-quality GBs. As a result, it not only mitigates the randomness introduced by *k*-means clustering but also enhances the model's robustness and generalization ability.

3.3 Algorithm and Its Time Complexity Analysis

The pseudocode of GB*k*NN-JGE is illustrated in Algorithm 1.

Algorithm 1: GB*k*NN-JGE

Input: Dataset D, parameters γ and *p*, query point x_{test}

Output: predicted label $\hat{y}$ for x_{test}

1: initialize list of base classifiers $\mathrm{H}\leftarrow\varnothing$;
2: **for** $i=1$ to m **do**
3: generate set of GBs on D using the strategy in Section 3.1 with parameters γ and *p*;
4: store base classifier H_i with its GB set;
5: **end for**
6: initialize ensemble score dictionary $S[c]=0$ for each class label $c\in\mathrm{C}$;

```
7:  foreach base classifier H_i ∈ H
8:      find nearest GB Ω_{U_i} to x_test ;
9:      y_i ←label of Ω_{U_i} and BQ(Ω_{U_i}) ←best quality of Ω_{U_i} ;
10:     S[y_i] = S[y_i] + BQ(Ω_{U_i}) ;
11: end for
12: return: ŷ by Eq. (9).
```

The time complexity of Algorithm 1 is analyzed as follows. Let n denote the number of instances in the dataset, and let g represent the approximate number of generated granular balls (GBs) per base classifier after line 3, where $g \ll n$. According to Section 3.1, Steps 1–3 have a time complexity of $O(n \log n)$; Step 4 has a time complexity of $O(ng)$; and Steps 5–6 have a time complexity of $O(g)$. Therefore, the time complexity of line 3 is approximately $O(n \log n + gn + g)$ [10]. Since the number of base classifiers is M, the total time complexity for lines 2–5 is $O(mn \log n + mkn + mk)$. For lines 7–11, the time complexity is $O(mk)$. Hence, the overall time complexity of Algorithm 1 is approximately $O(mn \log n + mkn + mk + mk)$.

4. Experimental Results and Analysis

In this section, the performance of GB-POJG is evaluated through several numerical experiments. The experiments were conducted on a PC equipped with an Intel Core Ultra 7 155H CPU and 32 GB of RAM. The software environment was Python 3.12. Table 1 provides detailed information about the datasets used in experiments. All the datasets are from the UCI Machine Learning Repository[11].

Accuracy is adopted as the primary evaluation metric in this study, as it is a classic and widely used indicator in classification tasks. Specifically, test accuracy is defined as the ratio of correctly predicted test instances to the total number of test instances. In this paper, the reported test accuracy corresponds to the average result over ten repetitions of 10-fold cross-validation, providing a robust estimate of model performance.

Table 1: The Details of Datasets

IDs	Datasets	#Instances	#Dimensions	#Classes
D1	Algerian Forest Fires	122	13	2
D2	Iris	150	4	3
D3	Speaker Accent Recognition	329	12	6
D4	Wholesale Customers	440	7	2
D5	Optical Recognition of Handwritten Digits	5620	64	10
D6	Website	1353	9	3
D7	Iranian Churn	3150	13	2
D8	Electrical Grid	10000	13	2
D9	Yeast	58509	48	11

4.1 Comparison between Different Methods

The proposed method is compared with the original GBkNN method and four classic machine learning classifiers, including kNN, SVM, Naive Bayes and Decision Tree. For GBkNN-JGE, the number of base classifiers is set to 30, and the remaining parameters are listed in Table 2..

Table 2: Parameter Settings

	D1	D2	D3	D4	D5	D6	D7	D8	D9
γ	0.9	0.8	0.1	0.9	0	0.8	0.9	0.3	1
p	1	1	0.95	1	1	0.9	1	0.8	0.9

The experimental results is presented in Table 3.

Table 3: Experimental Results on nine Datasets

Datasets	GB*k*NN-JGE	GB*k*NN	*k*NN	SVM	Naive Bayes	Decision Tree
D1	0.902	0.896	0.872	0.516	0.848	0.837
D2	0.973	0.932	0.965	0.966	0.953	0.944
D3	0.833	0.812	0.801	0.674	0.577	0.671
D4	0.827	0.818	0.796	0.798	0.577	0.671
D5	0.987	0.973	0.887	0.98	0.791	0.906
D6	0.86	0.825	0.774	0.772	0.718	0.786
D7	0.865	0.891	0.857	0.843	0.733	0.941
D8	0.784	0.78	0.79	0.891	0.879	0.904
D9	0.798	0.777	0.731	0.701	0.732	0.785
Avg(ACC)	0.870	0.856	0.83	0.783	0.756	0.827
Avg(Rank)	1.67	3	3.89	4.11	5	3.33

To compare the proposed algorithm with several baseline algorithms across multiple datasets, the non-parametric Friedman test and the Nemenyi test are adopted. Specifically, suppose there are h algorithms evaluated on N datasets. For each dataset, algorithms are ranked based on their test accuracy. Then the average rank is computed for each algorithm across all datasets. If all algorithms perform similarly, their average ranks should not differ significantly. The Friedman test is defined as:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2};\ \chi_F^2 = \frac{12N}{h(h+1)}\sum_{j=1}^{h}\left(R_j - \frac{h+1}{2}\right)^2 \tag{11}$$

If F_F exceeds the critical value, the null hypothesis that "all algorithms perform equally" is rejected. On rejecting the null hypothesis, the Nemenyi test is conducted to perform pairwise comparisons between algorithms. The critical difference (CD) is defined as: $CD = q_\alpha\sqrt{h(h+1)/6N}$. If the rank difference between two algorithms exceeds the CD, the performance difference between them is considered statistically significant.

For accuracy, based on Table 2, χ_F^2 and F_F are calculated as 16.4921 and 4.6821. At a significance level of α=0.05, the critical value is 2.422, leading to the rejectionof the null hypothesis. Furthermore, CD is calculated as 1.777 and the results of the Nemenyi test is displayed in Fig.2. The post-hoc analysis reveals that GB*k*NN-JGE demonstrates statistically significant performance differences when compared to Naive Bayes, SVM, and KNN models.

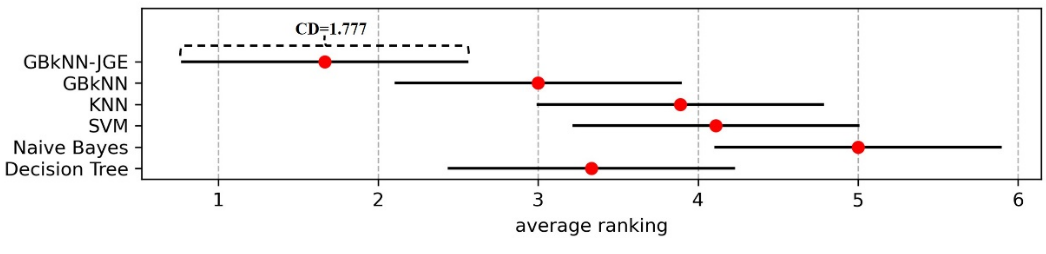

Fig.2: Nemenyi test with six algorithms on nine publicly available datasets

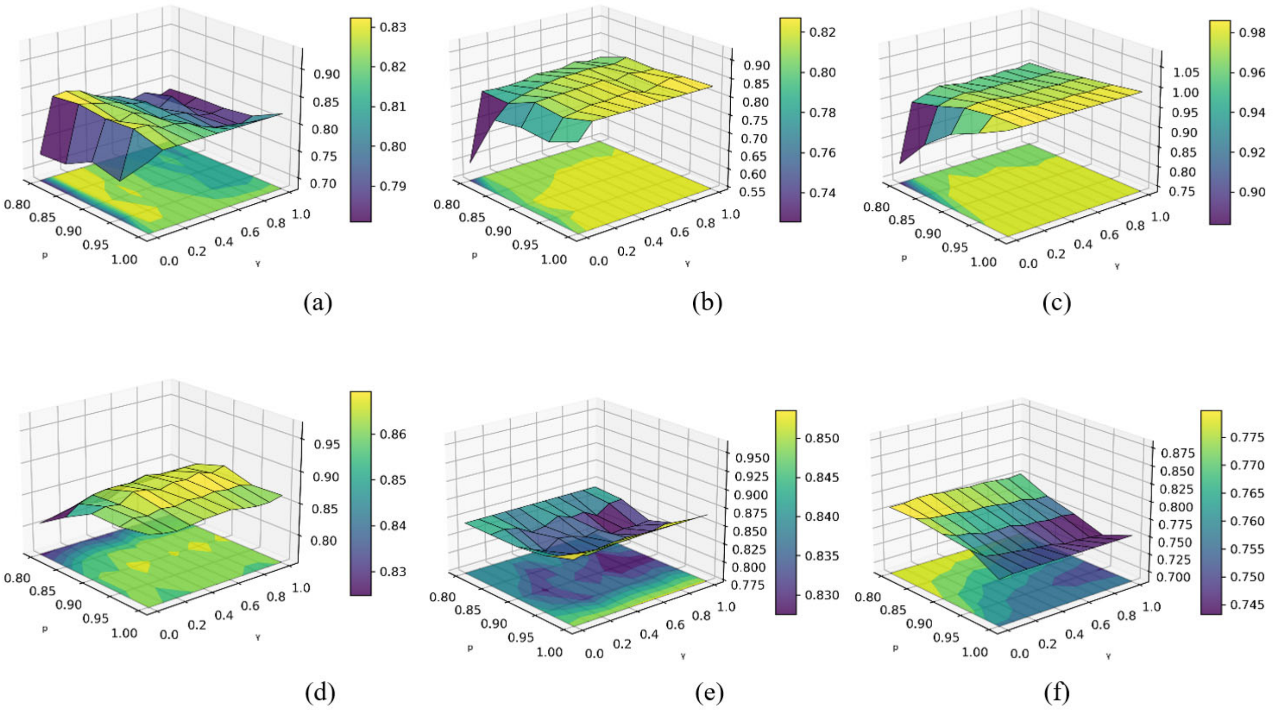

Fig. 3: Performance of GBkNN-JGE under different combinations of γ and p. (a)-(f) D3-D8.

In this section, the performance of GBkNN-JGE is further analyzed under various combinations of γ and p. Specifically, experiments are conducted on six selected datasets D3–D8. The value of γ is varied from 0 to 1 with a step size of 0.1, while the value of p is adjusted from 0.8 to 1 with a step size of 0.05. The results are presented in Fig. 2, which shows the accuracy for different combinations of γ and p.

From Fig. 3(a)-(f), it is evident that, for most datasets, accuracy remains relatively stable when γ is in the range [0.1,0.7], suggesting GBkNN-JGE is not overly sensitive to this parameter in that interval. While extreme values tend to lead to performance fluctuations, sometimes lower than the mid-range values. Additionally, moderate increases in p typically improve accuracy. However, when p reaches 1, the marginal gain becomes negligible or even negative for some datasets.

4.2 Ablation Experiments

In this section, ablation experiments are conducted on datasets D1–D9. Specifically, GBkNN-JGE is compared with five degenerated versions, described as follows: 1) A: replaces the quality-maximization-based weighted voting with a simple majority vote of the base classifiers' predictions for the final ensemble decision; 2) B: removes the ensemble learning component from GBkNN-JGE; 3) C: replaces the GB generation method with the one proposed in [7]; 4) D: replaces the GB generation method with the one proposed in [7] and replaces the weighted voting with majority voting based on the base classifiers' outputs;5) E: based on variant C, further removes the ensemble learning component.

Table 4: Results of Ablation Experiments

Datasets	GBkNN-JGE	A	B	C	D	E
D1	**0.897**	0.894	0.872	0.887	0.886	0.877
D2	**0.951**	**0.951**	0.932	0.945	0.949	0.949
D3	**0.816**	0.813	0.789	**0.816**	0.812	0.793
D4	**0.798**	0.789	0.781	0.795	0.792	0.782
D5	**0.989**	0.988	0.983	**0.989**	**0.989**	0.984
D6	**0.871**	0.867	0.844	**0.871**	0.863	0.843
D7	**0.862**	**0.862**	0.853	**0.862**	0.859	0.851

D8	**0.754**	0.739	0.724	0.747	0.737	0.725
D9	**0.798**	**0.798**	0.771	0.783	0.781	0.777
Average	**0.859**	0.855	0.838	0.855	0.852	0.842

The classification results are evaluated by accuracy, as shown in Table 4. It is observed that GBkNN-JGE achieves the highest accuracy on nine datasets. Moreover, the results in Table 3 demonstrate that ensemble learning outperforms individual classifiers. This improvement is primarily due to the ensemble strategy, which generates granules by repeatedly applying 2-means clustering, thereby mitigating the instability associated with a single run of 2-means clustering. In addition, the quality-maximization-based weighted voting yields better performance compared to majority voting based on the outputs of base classifiers.

5. Conclusion

In this article, a novel and stable granular classifier framework, termed GBkNN-JGE, is proposed for classification tasks. Compared with existing GB-based classifiers, GBkNN-JGE demonstrates the following advantages: 1) it employs a comprehensive metric to assess GB quality and incorporates a threshold-based strategy with the principle of maximizing overall quality; 2) An ensemble learning method is introduced to mitigate the instability caused by the randomness of k-means clustering in GB generation, ensuring more stable and reliable classification outcomes.

In future work, it would be valuable to explore adaptive parameter selection methods for determining optimal combinations of γ and p values. Additionally, the ensemble learning component could be further explored such as determining the optimal number of base classifiers and designing alternative ensemble strategies.

References

[1] Trabelsi, A., Elouedi, Z. and Lefevre, E., 2019. Decision tree classifiers for evidential attribute values and class labels. Fuzzy Sets and Systems, 366, pp.46-62.

[2] Cristianini, N. and Shawe-Taylor, J., 2000. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

[3] Rigatti, S.J., 2017. Random forest. Journal of Insurance Medicine, 47(1), pp.31-39.

[4] Regression, N., 1992. An Introduction to Kernel and Nearest-Neighbor. The American Statistician, 46(3), pp.175-185.

[5] Huang, D.S., 1996. Systematic theory of neural networks for pattern recognition. Publishing House of Electronic Industry of China, Beijing, 201.

[6] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. and Lopez, A., 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, pp.189-215.

[7] Xia, S., Liu, Y., Ding, X., Wang, G., Yu, H. and Luo, Y., 2019. Granular ball computing classifiers for efficient, scalable and robust learning. Information Sciences, 483, pp.136-152.

[8] Lang, G., Zhao, L., Miao, D. and Ding, W., 2025. Granular-Ball Computing Based Fuzzy Twin Support Vector Machine for Pattern Classification. IEEE Transactions on Fuzzy Systems.

[9] Xia, S., Dai, X., Wang, G., Gao, X. and Giem, E., 2022. An efficient and adaptive granular-ball generation method in classification problem. IEEE Transactions on Neural Networks and Learning Systems, 35(4), pp.5319-5331.

[10] Jia, Z., Zhang, Z. and Pedrycz, W., 2025. Generation of Granular-Balls for Clustering Based on the Principle of Justifiable Granularity. IEEE Transactions on Cybernetics.

[11] K. Bache and M. Lichman, 2017, "UCI machine learning repository," Datasets. [Online]. Available: http://archive.ics.uci.edu/ml