Research on Decision Tree Classification Algorithm Based on K-Nearest Neighbor Algorithms Guidance

Jianmei Chen, Xiaojun Ding*
School of Computer Science and Engineering, Yulin Normal University, Yulin, Guangxi, China
* *Corresponding Author.*

Abstract: Decision trees and k-nearest neighbor algorithms are classic classification methods in machine learning. Decision trees clearly display classification logic in a tree structure and are highly interpretable, but they are prone to overfitting in high-dimensional data and ignoring local details; k-nearest neighbors capture features through voting, which is suitable for local patterns but lacks global grasp. The KNN_DT algorithm innovatively combines the advantages of both, with both local flexibility and global control. This report deeply analyzes the core ideas and principles of the KNN_DT algorithm, aiming to provide a solid theoretical foundation and reference for its research and application, and to promote efficient and accurate data processing and application transformation in various industries.

Keywords: Decision Trees; K-Nearest Neighbor Algorithms; Local Information; Global Information

1. Introduction

In today's booming digital age, machine learning has become a key driving force in advancing various fields, with classification algorithms, as the fundamental core, receiving significant attention. Decision trees and the k-Nearest Neighbor (KNN) algorithm are considered classics and are widely applied in practice.

The decision tree follows the tree-like construction logic, breaking down the complex classification process into a series of feature testing links, advancing from the root node layer by layer to the leaf node to give the classification result. Its clear structure gives excellent interpretability and facilitates insight into data classification rules [1]. In contrast, the k-nearest neighbor algorithm adheres to the concept of local majority voting, takes the test sample as the core, explores the surrounding k "neighbor" categories in the data space, and determines the final attribution based on this. It can accurately capture local structural features and show strong classification performance in local areas [2].

Although decision trees and k-nearest neighbor algorithms are classic, they have obvious shortcomings. Decision trees are prone to overfitting when processing high-dimensional data, focusing on global information to divide nodes, often ignoring local structures, resulting in insufficient generalization ability; k-nearest neighbor algorithms focus on local areas, rely on neighbor sample decisions, and are difficult to grasp global structures. They are inefficient when facing large-scale complex data sets, and require the storage of all training data, and have slow query and classification speeds [3]. With the advent of the big data era, the amount and complexity of data have increased dramatically, and a single algorithm is difficult to meet the needs. It has become a top priority to explore algorithm fusion or develop innovative comprehensive algorithms. Integrating the advantages of different algorithms can not only improve the processing capabilities of large-scale complex data sets, but also provide more robust and flexible solutions for real-world problems, helping various industries to achieve efficient and accurate data processing and application transformation.

The k-nearest neighbor-guided decision tree (KNN_DT) algorithm innovatively combines the local flexibility of k-nearest neighbors and the global control of decision trees, inherits the advantages of k-nearest neighbors in capturing local structures and the systematic and comprehensive nature of decision trees in building classification models, and provides an efficient and novel solution for complex

problems, driving various industries to break through the waves of digitalization. This report will delve deeply into the core ideas and algorithm principles of KNN_DT, providing a theoretical foundation and reference for the research and application of this innovative algorithm.

2. Related Work

In the continuous evolution of machine learning, decision trees and the KNN algorithm, as classic classification methods, have received much attention. The academic community has widely carried out research, deeply exploring the internal potential of these two algorithms, and making every effort to overcome their limitations. On the one hand, efforts are focused on improving the accuracy of the algorithms, enhancing their generalization performance, and optimizing computational efficiency. On the other hand, the scope of application is expanded to adapt to the increase in the scale and complexity of data. Researchers continue to explore ways to integrate the strong interpretability of decision trees and the ability of KNN to capture local features, laying a solid foundation for solving real-world problems and producing robust and efficient solutions.

2.1 Research Progress on Decision Trees and K-Nearest Neighbors

In the field of machine learning, decision tree algorithms and KNN algorithms have their own characteristics and have a profound impact on the development of data analysis and pattern recognition.

Decision tree algorithms have gained a firm foothold with their intuitive tree structure and excellent interpretability. In the early days, the ID3 algorithm selected features based on information gain based on Shannon entropy to promote node division [4]; the C4.5 algorithm optimized the former and was able to handle continuous values and fill in the gaps in missing value processing [5]; the CART algorithm used the Gini coefficient to recursively partition the binary tree, which was compatible with classification and regression, laying the foundation for its development [6]. However, traditional decision trees have many drawbacks in their application expansion, especially overfitting. The uncontrolled tree depth leads to poor generalization and weak performance on new data, which limits their practical application. Therefore, controlling complexity and overcoming overfitting became the focus, and strategies such as pruning technology and ensemble learning came into being. For example, random forests use self-service sampling to build multiple decision trees to reduce the risk of overfitting and improve accuracy and robustness [7]; gradient boosting trees use serial training and error correction optimization, and have performed well in competitions and practical applications [8]. However, both algorithms are strong in global optimization and weak in local mining. When faced with tasks dominated by local features, it is difficult to mine the full value of data. In this context, integrating global and local features and improving the adaptability of complex data have become the forefront, giving rise to new algorithms such as K-nearest neighbor-guided decision trees to make up for the shortcomings.

Similarly, based on the principle of "nearest neighbor voting", KNN algorithm can perform classification and regression tasks without the need for complex training. In theoretical research, researchers have delved deeply into distance metrics. In addition to Euclidean distance and Manhattan distance [9], they have introduced Mahalanobis distance and others to adapt to diverse scenarios. However, there are numerous obstacles in its development. Firstly, high-dimensional data gives rise to the "curse of dimensionality". As the number of feature dimensions increases, it becomes difficult to distinguish the distances between samples, and the concept of "nearest neighbors" becomes ambiguous, leading to a decline in accuracy. When dealing with large-scale training sets, the exhaustive calculation of distances causes a sharp drop in query efficiency, consuming a large amount of time and computational resources. Moreover, it is difficult to select an appropriate K value. An inappropriate K value may lead to sensitivity to noise or underfitting, and the process of finding the optimal value is time-consuming. To address these dilemmas, approximate nearest neighbor search algorithms and data structures such as KDTree [10] and BallTree [11] have achieved some success in improving query efficiency. However, in the face of complex high-dimensional data distributions, the "curse

of dimensionality" still exists, and there is an urgent need to explore a universal optimization path to ensure its stable development. Both the decision tree and the K-Nearest Neighbors algorithm have their own advantages and disadvantages. The exploration of their integration is of great significance and is expected to open up a new path for the development of machine learning.

2.2 Exploration of the Combination of Decision Tree and K-nearest Neighbor Algorithm

In the field of machine learning, the decision tree and KNN algorithm have their unique advantages and significant limitations, which has led to the organic combination of the two becoming a hot trend. The core goal is to integrate the characteristics of both parties and improve the model performance in all aspects.

Some scholars have tried to introduce the concept of K-nearest neighbor in the decision tree construction process [12]. The specific practice is to use the K-nearest neighbor algorithm to perform classification operations at the leaf nodes of the decision tree. This method has outstanding advantages in situations where the number of samples is small and the category distribution is uneven. With the characteristics of K-nearest neighbor's flexible adaptation to data distribution, it can effectively fill the shortcomings of decision trees in local fine classification, so that the classification effect is improved to a certain extent.

However, this initial combination mode is relatively simple and limited to supplementing the decision tree with K-nearest neighbors at the end, lacking fine integration and deep optimization of global and local information [2]. Subsequent research can consider integrating the local sensitivity of the KNN algorithm into the feature selection or split rule formulation of the decision tree, so that the two can work closely together at key nodes; on the other hand, design a new framework to enable the decision tree and K-nearest neighbor to support each other at different levels and form an organic whole. Through these in-depth strategies, the accuracy and robustness of the model will be improved, and complex data structures and distributions can be properly handled.

At the same time, some research has focused on the feature selection stage. By taking advantage of the sensitivity of the KNN algorithm to the local structure of data, the importance of features is re-evaluated. That is, by observing how specific features affect the consistency of the classes of samples in the local neighborhood, more discriminative features are selected for the decision tree division process. However, this method still needs to be improved in terms of coordinating global division and local optimization, especially in enabling the model to balance effective macroscopic classification and fine-grained microscopic processing of local information.

The KNN_DT algorithm proposed in this study shows outstanding innovation. It abandons the simple patching or single-stage fusion of traditional decision trees, and instead deeply integrates the local information perception ability of the K nearest neighbor algorithm with the global partitioning strategy of the decision tree, ultimately achieving higher classification performance on complex data sets.

3. KNN_DT Method

3.1 Detailed Explanation of the Core Idea of KNN_DT Algorithm

In the process of machine learning pursuing accuracy and efficiency, KNN_DT algorithm brings innovative solutions to complex data classification. The key lies in the integration of decision tree global division and k nearest neighbor local perception ability:

Dual perspective fusion: This algorithm not only continues the decision tree's advantage of controlling the global structure, but also reduces the impurity of sub-nodes and regularizes data by partitioning, and introduces k-nearest neighbor local neighborhood information. When evaluating the partition, it takes into account the improvement of data purity after global segmentation and the local guarantee that the data point and the K nearest neighbors belong to the same sub-node, so as to take into account both macro trends and micro structures, improve the adaptability and accuracy of the model, and deal with practical problems.

Dual perspective improves performance Reason: Globally, the decision tree uses

indicators such as information gain to roughly divide data and screen features, laying the foundation for fine classification; the local perspective makes up for its ignoring details. In the case of local heterogeneous sample aggregation, the close neighbor relationship is accurately divided to prevent overfitting and improve generalization. The fusion of the two allows the algorithm to switch reliance according to the data scenario, evenly emphasize the decision tree, and play the guiding role of neighborhood information in complex local areas.

Measurement of Local Neighborhood Information: For each data point in a node, find its K nearest neighbors. After the division, count the number of neighbors that belong to the same child node. The higher this number, the better the local preservation. Additionally, based on the categories of the neighbors and their distribution after the division, use a probability model to estimate the probability of local consistency. This approach replaces simple counting and provides a more refined evaluation.

Information Integration Strategy: Set up a comprehensive evaluation function TotalScore = α * GlobalGain + (1 - α) * LocalConsistencyScore. Here, GlobalGain represents the global gain of the decision tree, LocalConsistencyScore is the local score, and α is a hyperparameter used to adjust the balance. When α = 1, it degenerates into a standard decision tree, and when α is close to 0, more emphasis is placed on local maintenance.

KNN_DT Algorithm:

1. Initialization: Start from the root node, which contains all the training data.
2. Recursive Node Partitioning: Perform the following operations on the current node:
   a) Stopping Condition Check: If the node meets the stopping condition, mark the node as a leaf node. The predicted value of the leaf node can be the majority class of the samples in that node.
      i. Finding the Optimal Partition: Traverse all features j and all possible partition points v of that feature:
         1. Candidate Partition: According to feature j and partition point v, divide the dataset D_node of the current node into two subsets: D_left (feature j <= v) and D_right (feature j > v).
         2. Calculate Global Gain: Calculate the traditional impurity reduction brought by this partition.
         3. Calculate Local Consistency: For each data point p in the current node D_node:
            a) Find the K nearest neighbors N_k(p) of p within the entire training set or the current node D_node.
            b) Evaluate the local consistency of this partition for p. The basic method is to calculate how many of the K neighbors of p are partitioned into the same child node as p.
            c) Aggregate the local consistency scores of all points p to obtain the LocalConsistencyScore of this candidate partition.
         4. Combine Evaluation Criteria: Design a new evaluation function that combines global gain and local consistency to evaluate the quality of the partition.
         5. Select the Optimal: Select the feature j and partition point v that maximize the TotalScore as the optimal partition for the current node.
      b) Create Child Nodes: Use the optimal partition (j, v) to split the data of the current node into two new child nodes.
      c) Recursion: Repeat step 2 for the generated child nodes.

3.2 Algorithm Design and Implementation

In the field of machine learning, feature selection is crucial to building efficient models, and it is related to model performance and interpretability. The KNN_DT algorithm combines the advantages of decision trees and k-nearest neighbors, and its feature selection is unique:

3.2.1 Feature selection based on local neighborhood information

Local neighborhood information acquisition: Feature selection introduces the k-nearest neighbor perception of local information. For each data point of the current node, find the k nearest neighbors according to the predefined distance metric (such as Euclidean distance), such as using the Euclidean distance formula to calculate the distance between data points and sort and select the first k.

Evaluation indicators and methods: When evaluating the partition, the relationship between the data point and the K nearest neighbors is considered. A simple way is to calculate the number of child nodes that the K

neighbors of the data point have in common with it as the local consistency score.

Feature Selection Decision: Combine the local indicators with the global indicators of traditional decision trees (such as information gain and Gini impurity). A commonly used linearly weighted comprehensive evaluation function is TotalScore = α * GlobalGain + (1 - α) * LocalConsistencyScore. Here, α is used to adjust the balance. Select the feature that maximizes the TotalScore to take both global organization and local structure into account, prevent local optima, and improve the generalization ability.

3.2.2 Global information evaluation

Traditional decision tree node partitioning uses global - information - based indicators such as information gain and Gini gain. Taking information gain as an example, the effect of feature partitioning is measured by calculating the change in entropy before and after the partition. A larger information gain indicates a better partition, which can reduce the impurity of child nodes.

3.2.3 Comprehensive evaluation and node division KNN_DT algorithm sets a new evaluation function TotalScore = α * GlobalGain + (1 - α) * LocalConsistencyScore to balance global and local information, and the importance of α. The actual division traverses the features and division points, and selects the one with the largest TotalScore as the best division, using dual information sources to improve fitting and generalization capabilities.

Through a comprehensive analysis of the KNN_DT algorithm, its innovation lies in the organic integration of the advantages of the decision tree and the KNN algorithm, directly addressing and attempting to overcome the limitations of traditional classification algorithms. However, the value of the algorithm needs to be verified through practice. Therefore, a series of experiments in Chapter 4 have been carefully prepared, aiming to confirm the effectiveness and superiority of the KNN_DT algorithm.

4. Experiments

The experimental data set is from the UCI public database. We selected very representative data sets covering a wide range of fields,

including heart disease data sets, ionosphere data sets, Wisconsin breast cancer diagnosis data sets, hepatitis data sets, and automobile evaluation data sets related to comprehensive evaluation, as well as mushroom data sets that may involve biological related fields. These rich and diverse data sets cover different data distributions, feature dimensions, and sample sizes, and can fully test the adaptability and generalization capabilities of the KNN_DT algorithm. The experimental design is scientific and rigorous, and the KNN_DT algorithm is compared with the traditional decision tree algorithm, the K-nearest neighbor algorithm, and other mainstream classification algorithms. To ensure fairness, all algorithms are run under the same hardware environment and data preprocessing process, and the k-fold cross validation is used to repeat the experiment multiple times to obtain reliable performance evaluation indicators.

Evaluation indicators take into account both classification accuracy and operational efficiency. Indicators such as classification accuracy, recall rate, and F1 value are used to measure the ability of the algorithm to correctly classify samples; operational efficiency indicators such as computing time and memory usage evaluate the resource consumption of the algorithm. Taking large-scale medical imaging data processing as an example, fast and accurate diagnosis and classification are related to patient treatment and medical resource allocation, and comprehensive consideration of these indicators is very critical.

In order to further explore the performance of the KNN_DT algorithm under complex data structures, challenging data scenarios such as introducing noise data and simulating unbalanced data distribution are deliberately constructed. By observing its performance changes under harsh conditions, the stability and robustness of the algorithm are further verified.

This experiment aims to compare the performance of the KNN_DT algorithm and the CART algorithm under different parameter settings. By taking 0.5, 0.7, and 0.9 on 8 representative data sets, and taking k values of k = 5 and k = 7 for cross-validation, a series of results with reference value are obtained.

From the experimental results (Table 1), we can see that when k = 5: α = 0.5, KNN_DT

has better classification ability than CART on 4 data sets, the two are equal on 1 data set, and the accuracy is lower than CART on 3 data sets; α = 0.7, KNN_DT has better classification ability than CART on 6 data sets, and lower than CART on 2 data sets; α = 0.9, KNN_DT has better classification ability than CART on 6 data sets, the two are equal on 1 data set, and worse than CART on 1 data set. Overall, as α increases, the performance of the KNN_DT algorithm exceeds that of the CART algorithm on more data sets.

Table 1. Accuracy of k=5 Cross Validation

	A=0.5		A=0.7		A=0.9	
	KNN_DT	CART	KNN_DT	CART	KNN_DT	CART
zoo	0.8387	0.8710	**0.8710**	0.8065	**0.9032**	0.9
ionosphere	**0.9245**	0.9057	**0.9245**	0.8396	**0.8779**	0.8663
breast cancer	**0.9591**	0.9298	0.9298	0.9532	**0.9381**	0.9298
car evaluation	**0.8054**	0.8054	**0.8582**	0.8536	0.7938	0.8246
mushroom	**1**	1	**1**	0.9938	0**.9918**	**0.9918**
german credit	0.6667	0.72	**0.7433**	0.74	**0.6967**	0.69
spambase	0.7726	0.9102	0.8921	0.9073	**0.9001**	0.8993
spectf heart	**0.7654**	0.7654	**0.7901**	0.7654	**0.7160**	0.6913

Table 2. Accuracy of k=7 Cross Validation

	A=0.5		A=0.7		A=0.9	
	KNN_DT	CART	KNN_DT	CART	KNN_DT	CART
zoo	0.9677	0.9132	**0.9032**	0.9032	**0.9677**	0.9633
ionosphere	**0.8962**	0.8396	**0.9057**	0.8368	**0.9151**	0.9057
breast cancer	**0.9240**	0.9123	**0.9415**	**0.9415**	**0.9540**	0.9358
car evaluation	**0.8095**	0.8092	**0.8092**	0.8201	**0.8208**	0.8193
mushroom	**1**	0.9897	**1**	**1**	**0.9906**	**0.9906**
german credit	0.6867	0.7	**0.72**	0.67	**0.6867**	0.6833
spambase	0.7524	0.8986	0.9111	0.9066	**0.9188**	0.9109
spectf heart	**0.8395**	0.8272	**0.7901**	0.7407	**0.7531**	0.6667

From the experimental results (Table 2), it can be seen that when k = 7: α = 0.5, KNN_DT is better than CART in 5 data sets, and the accuracy of 3 data sets is lower than CART; α = 0.7, KNN_DT is better than CART in 5 data sets, 2 data sets are equivalent to CART, and 1 data set is lower than CART; α = 0.9, KNN_DT is better than CART in 7 data sets, and 1 data set is the same as CART. Overall, as α increases, the performance of the KNN_DT algorithm exceeds that of the CART algorithm on more data sets.

The KNN_DT algorithm aims to enhance the adaptability and accuracy of the model by integrating the global partitioning ability of the decision tree and the local perception ability of K-nearest neighbors. The experimental results demonstrate that the hyperparameters α and k have a significant impact on the performance of the algorithm. A adjusts the balance between global and local information, and different datasets exhibit varying sensitivities to α, with α having a more pronounced effect on complex datasets. K determines the scope of the local neighborhood and has a greater influence on datasets with uneven local density or the presence of noise. Compared with the traditional CART algorithm, the KNN_DT algorithm shows competitiveness on most datasets.

When α = 0.5, the weights of global information and local information are the same. At this time, the performance of the algorithm is more influenced by local information. When α = 0.7, the weight of global information exceeds that of local information, and the performance of the algorithm is more affected by global information. When α = 0.9, the impact of local information on the performance of the algorithm is relatively small.

5. Conclusion and Outlook

This study focuses on the KNN_DT algorithm, innovatively integrating k-nearest neighbor local perception and decision tree global partitioning strategy, bringing vitality to machine learning classification. It breaks the limitations of traditional decision trees,

integrates local considerations, takes into account both the whole and the local, has solid theory, enhances generalization, avoids local optimality, and can cope with complex data. This paper explains its core, feature selection, node partitioning and attaches pseudo code. Experiments show that compared with mainstream algorithms, it has excellent accuracy in multi-domain data sets, accurately handles complex data, has low misclassification rate, and saves resources by using tree structure in terms of computational efficiency. However, local neighborhood calculations are costly when encountering high data volume and dimensions, and it is planned to combine distributed computing and other technical optimizations in the future. Hyperparameter tuning currently relies on manual or simple grid search, and it is planned to be automated using Bayesian optimization in the future. The application has been involved in many fields, and it is expected to be further expanded in the face of emerging complex data, and the trend of integration with deep learning is gradually emerging. In the future, it will continue to empower and overcome difficult problems.

References

[1] Krall, M.A., A.V. Gundlapalli and M.H. Samore, Chapter 13 - Big Data and Population-Based Decision Support, in Clinical Decision Support (Second Edition), R.A. Greenes, R.A. Greenes^Editors. 2014, Academic Press: Oxford. p. 363-381.

[2] Zhao F, Zhang M, Zhou S, Lou Q. Detection of network security traffic anomalies based on machine learning KNN method. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023 2024; 1:209-18.

[3] Le, L., Y. Xie and V.V. Raghavan, KNN loss and deep KNN. Fundamenta informaticae, 2021. 182(2): p. 95-110.

[4] Deng Z, Loyher P, Lazarov T, Li L, Shen Z, et al. The nuclear factor ID3 endows macrophages with a potent anti-tumour activity. Nature 2024; 626:864-73.

[5] Febriani, S., Analisis Data Hasil Diagnosa Untuk Klasifikasi Gangguan Kepribadian Menggunakan Algoritma C4. 5. Jurnal Ilmu Data, 2022. 2(9).

[6] Agustin, R. and S. Defit, Perbandingan Algoritma CART dan C. 4 5 Pada Citra Tandan Buah Sawit Untuk Mengetahui Tingkat Kematangan Dalam Penentuan Harga. Jurnal KomtekInfo, 2024: p. 263-273.

[7] Del Río S, López V, Benítez JM, Herrera F. On the use of MapReduce for imbalanced big data using Random Forest. Inf Sci 2014; 285:112-37. 10.1016/j.ins.2014.03.043.

[8] Hashemizadeh A, Maaref A, Shateri M, Larestani A, Hemmati-Sarapardeh A. Experimental measurement and modeling of water-based drilling mud density using adaptive boosting decision tree, support vector machine, and K-nearest neighbors: A case study from the South Pars gas field. J Pet Sci Eng 2021; 207:109132.

[9] Ren J, Fort S, Liu J, Roy AG, Padhy S, et al., A simple fix to mahalanobis distance for improving near-ood detection. arXiv preprint arXiv:2106.09022, 2021.

[10] Gutiérrez, G., R. Torres-Avilés and M. Caniupán, cKd-tree: A Compact Kd-tree. IEEE Access, 2024. 12: p. 28666-28676.

[11] Zhang L, Wang G, Peng L, Peng W, Zhang J. Applying pareto frontier theory and ball tree algorithms to optimize growth boundaries for sustainable mountain cities. Journal of Urban Management, 2024.

[12] Dinesh, P., A.S. Vickram and P. Kalyanasundaram. Medical image prediction for diagnosis of breast cancer disease comparing the machine learning algorithms: SVM, KNN, logistic regression, random forest and decision tree to measure accuracy. in AIP Conference Proceedings. 2024: AIP Publishing.

