Comparative Analysis and Optimization of K-Nearest Neighbor and Decision Tree Classification Techniques

Abstract

This work attempts to compare the performance of two distinct classification techniques, K-Nearest Neighbors (KNN) and Decision Tree (DT), on two different health datasets: breast cancer and hepatitis. We found that KNN consistently demonstrated better accuracy over Decision Tree for the most of time on the given datasets, but DT showed excellent performance on feature minimization. We also investigated the influence of various hyperparameters and cost/distance functions on the performance of these models and the characteristics of their decision boundaries; then we developed several optimization techniques for each classification method, i.e. introducing KD Tree search algorithm into KNN classifier to reduce searching time, implementing pre/post-pruning methods for DT to decrease the chances of overfitting.

1. Introduction

Classification aims to categorize all instances in the datasets into distinct classes based on their attributes. This report explores two distinctive classification approaches, *K-Nearest Neighbors* and *Decision Tree*, examines their performance among different hyper-parameters on the tasks like offering basic diagnosis of the types of breast cancer tumors and hepatitis, and compares the strengths and weaknesses of these two models after multiple optimization strategies, for example, searching in *KD Tree* and *pre/post-pruning* of tree structured data, have been deployed.

1.1 Background Information

K-Nearest Neighbors is a non-parametric classification method which stores all input data and makes predictions based on *k* data points with minimum distances to the test data.

Decision Tree is a modelling approach via splitting training data through optimal tests into tree structured clusters represented by leaves, then making predictions.

KD Tree is a binary tree which partitions k-dimensional data points into tree-structured clusters.

Pre/Post-Pruning is a simplifying technique which prevents decision trees from overfitting. Pre-pruning works by stopping splitting early before the tree overfits the training data. In contrast, post-pruning reduces the depth of the tree and the number of leaves after the training process has been completed.

1.2 Task Description

In this work, we develop binary classification models to predict the types of breast cancer tumors: benign (Class 2) or malignant (Class 4), and the classes of hepatitis: live (Class 1) or die (Class 2). The given data were first sorted into training and testing categories. In both parts of the data, the labels were separated from the data and used for supervision. Besides the labels, instances in the breast cancer dataset contain 10 attributes and data in the hepatitis set include 19 attributes, and we attempted to utilize these features to make predictions. The characteristics of the data will be further detailed in Section 2.

We can in general divide our tasks in three parts. First, to suppress noise, we preprocessed the raw dataset acquired from the source website and collected basic statistics to get some insights about the data distribution. Second, we implemented both models, KNN and decision tree, from scratch as well as some special functions used for additional experiments. Last, we ran the basic experiments on our models, and compared the accuracy of both cross-validated models on each dataset. In addition, we examined how the value of hyperparameters in both models will affect their performance. For KNN, we tried different K values and analyzed its effects on the training data and testing data accuracy. For the decision tree model, we tested how maximum depth and minimum number of data in each leaf of the decision tree will influence the predictive accuracy of the model on the test data. And we tried different cost functions and compared how they will impact the accuracy. We also visualized the decision boundaries of each model and showed their differences. Besides the basic experiments above, we developed a few optimization techniques to boost the performance of our classifiers. For KNN, to obtain the k nearest neighbors of the testing data, we normalized

the values of all features, implemented inverse distance weighting voting method, and developed a more efficient searching algorithm using KD-Tree. For decision tree, two additional pruning methods had been implemented: one of them is pre-pruning, which stops splitting when there would be no significant differences (less than 10-20%) in cost between the parent node and its two leaves nodes, and the other one is post-pruning, which greedily merges the leaves of the decision tree with their parents until it hurts the accuracy of the model on the validation set.

1.3 Related Work

Many previous works using the same dataset have also explored various approaches to optimize the performance of decision trees and KNN. In particular, Floriana, Donato, and Giovanni conducted a comparative analysis of six methods for pruning decision trees, and all of those pruning algorithms consistently reach approximately 80% accuracy on the hepatitis dataset.

Also, David B. Skalak discussed the advantages and limitations of various composite nearest neighbor classifiers using the breast cancer data. He also demonstrated that a standard KNN classifier would be able to achieve 90+% accuracy on the breast cancer dataset.

2. Data and setup

In this section, we provide the context on the breast cancer and hepatitis datasets and then describe general data-preprocessing methods that are common to all our approaches.

2.1 Dataset Source

Breast cancer and hepatitis datasets come from UCI's online machine learning repository available for access to personal and non-commercial use.

2.2 Preprocessing

The preprocessing phase consists of eliminating all instances containing missing or malformed value (entries filled by "?"), which will not be considered in the experiments; also, the attribute "ID" in the breast cancer dataset has been dropped since it's simply an index and will not offer any insight into the tumors; then we collected basic statistics about the distribution of certain features and labels to address potential bias.

2.3 Information

Besides the labels, instances in the breast cancer dataset contain 10 attributes and all of them are integers from 1 to 10 (except ID, which is a 7-digit number). In contrast, instances in the hepatitis dataset include 19 attributes in addition to the labels. Some attributes, sex for example, are binary (yes/no) features and have been encoded as 1 and 2; some attributes, such as age, are integers; the rest of attributes, like ALBUMIN, are continuous.

After the cleaning process, the hepatitis contains 80 instances: 47(58.75%) of them are from class 1, and 33(41.25%) of them are from class 2. The breast cancer dataset has 683 instances: 444(65.01%) of them are from class 2, and 239(34.99%) of them are from class 4. This statistics shows that both datasets are in some degree imbalanced, and the breast cancer dataset is more imbalanced than the hepatitis dataset. Besides the imbalance in the labels, the distribution of some attributes is also imbalanced. For example, in the hepatitis dataset, 69(86.25%) cases are males, but only 11(13.75%) cases are females. This will introduce undesirable bias and ethical concerns which will be further discussed later.

2.4 Ethical Concerns

As previously mentioned, both datasets are somewhat imbalanced, so it's very likely to introduce inductive bias into our models. For instance, 65.01% of breast cancer tumors in the dataset are benign, so the model may be more adapted to predict benign tumors rather than malignant ones, potentially raising the rate of misdiagnosis if the model was deployed in clinical practice. Plus, in the hepatitis dataset, only 13.75% cases come from female patients. It's possible to cause inductive bias so that the model will be less capable of accurately classifying hepatitis for female patients, which discriminates against females. Therefore, we should carefully address the potential bias as much as possible.

3. Result

3.1 Accuracy: KNN vs Decision Tree

Via 5-fold cross validation, the hyperparameters, distance functions, scaling, and cost functions of both models have been optimized, and will be further detailed later. 80% of data was used for training and validation, and the rest was for testing. To summarize, both models achieve similar accuracy on the breast cancer dataset, but KNN classifier is more predictive on the hepatitis dataset.

Hyperparameters and settings

For the breast cancer dataset:
K = 5
Voting: Inverse distance weighting
Distance function: Manhattan distance
Maximum depth = 3
Minimum number of data in each leaf (N) = 5
Cost function = misclassification rate

For the hepatitis dataset:
K = 9
Voting: Inverse distance weighting
Distance function: Manhattan distance
Maximum depth = 4
Minimum number of data in each leaf (N) = 5
Cost function = misclassification rate

Accuracy(%)	KNN	Decision Tree
Breast Cancer	**95.37±1.87%**	**95.60±2.03%**
Hepatitis	**86.68±5.41%**	**80.57±8.08%**

3.2 K value of KNN classifier

The following experiment was conducted on the breast cancer dataset using Manhattan distance function and distance inverse weighting voting.

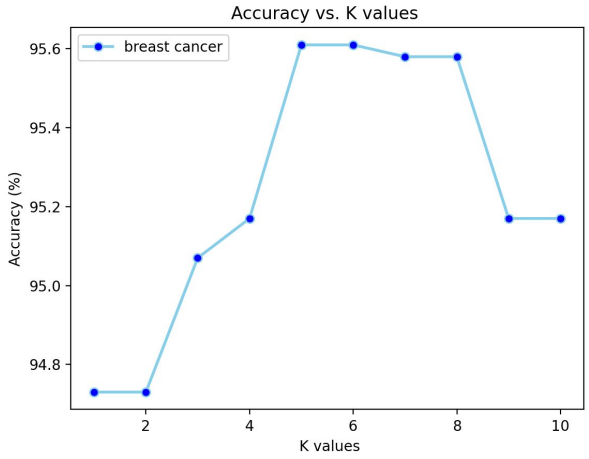


Figure 1: Accuracy vs. K values

The plot above shows that K=5 is the minimum value which reaches the optimal accuracy. When K<5, the accuracy is lower but still increasing, so the model is overfitting. Then the accuracy drops after K exceeds the optimal value, implying the classifier underfits. An intermediate value of K optimizes the accuracy of the KNN classifier.

3.3 Maximum depth

This experiment used maximum depth as the only pre/post-pruning method, so the minimum number of data in each leaf is set to 1, and the cost function is misclassification rate. The hepatitis dataset was used.

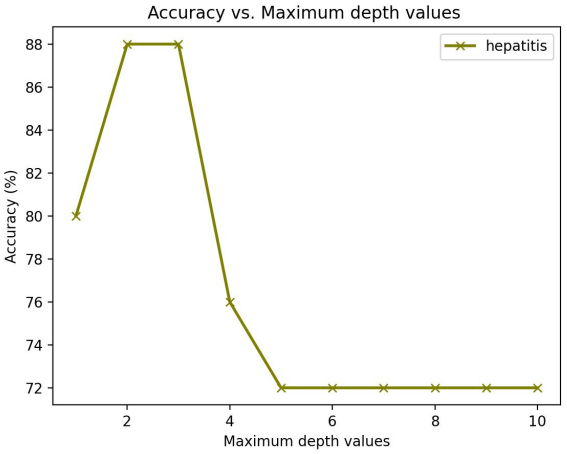


Figure 2: Accuracy vs. Maximum Depth

This figure demonstrates that the accuracy came to its climax when the maximum depth was between 2 and 3. Before that, the decision tree model has a lower accuracy, which is a sign of underfitting. After the climax, the accuracy drastically decreased and finally leveled off to a lower accuracy, indicating that the decision tree overfitted and failed to impact the accuracy anymore.

3.4 Distance/Cost Functions

3.4.1 Distance Functions

Distance functions quantify the level of similarity between data points. In KNN, for a given test data point, k nearest points around the test data were chosen and weighted to predict the label for the test data. In this report, we compared the performance of two distance functions, Euclidean distance and Manhattan distance, and evaluated their test accuracy.

Euclidean distance:

$$D_{Euclidean}(x, x') = \sqrt{\sum_{d=1}^{D} (x_d - x_d')^2}$$

Manhattan distance

$$D_{Manhattan}(x, x') = \sum_{d=1}^{D} |x_d - x_d'|$$

Two KNN classifiers using these two distance functions were tested on different datasets.
In this experiment, K=5, and voting is weighted by inverse distance.

Accuracy(%)	Euclidean	Manhattan
Breast Cancer	**94.60±3.69%**	**94.91±3.99%**
Hepatitis	**85.22±5.88%**	**87.59±5.65%**

The result above supports the conclusion that different distance functions indeed have different influences on the performance of the KNN

classifiers, and for these two datasets, Manhattan distance is slightly better for KNN classifiers.

3.4.2 Cost Functions

In decision trees, cost functions can measure how well the candidate tests separate and purify data of different classes, which helps us to pick the most essential features, minimize the number of features used, and make predictions based on in which clusters the test data fall. In this report, we analyzed the performance of three cost functions: gini index, entropy, and misclassification rate.

Gini Index/Gini Impurity

$$cost(R_k, D) = \sum_{c=1}^{C} p(c)(1-p(c)) = 1 - \sum_{c=1}^{C} p(c)^2$$

Entropy

$$cost(R_k, D) = H(y) = -\sum_{c=1}^{C} p(y=c) \cdot log\, p(y=c)$$

Misclassification Rate

$$cost(R_k, D) = \frac{1}{Nk} \sum_{x^{(n)} \in R_k} II(y^{(n)} \neq w_k) = 1 - p_w(w_k)$$

where $w_k = argmax_c\, p_c(c)$

Three decision trees, trained with different cost functions above, were tested on both datasets. In this experiment, maximum depth=4, N=5.

Accuracy(%)	Gini	Entropy	Misclassification
Breast Cancer	**94.52 ±1.72%**	**94.83 ±1.97%**	**95.09 ±2.03%**
Hepatitis	**81.29 ±8.97%**	**80.67 ±8.46%**	**81.67 ±8.09%**

This result above shows that cost functions indeed impact the accuracy of decision tree classifiers, and misclassification rate outperforms other cost functions a little, which illustrates that a more complex cost function does not necessarily lead to better accuracy. Sometimes a simpler approach can both make learning easier and get satisfactory results.

3.5 Characteristics of Decision Boundary

In this part we focus on the characteristics of the decision boundaries of KNN and decision tree classifiers, and observe their differences. Here we used the breast cancer dataset for illustration.

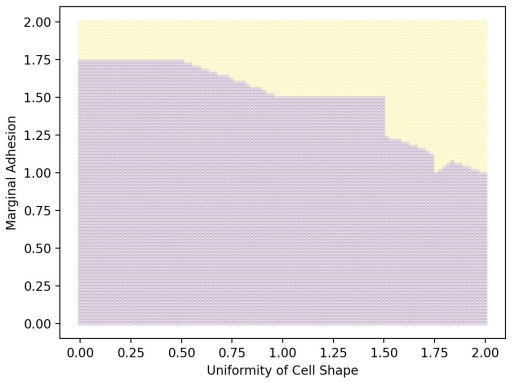


Figure 4: A Decision Boundary of KNN Classifier. Instances in the yellow area will be classified as malignant tumors. Others will be predicted as benign ones.

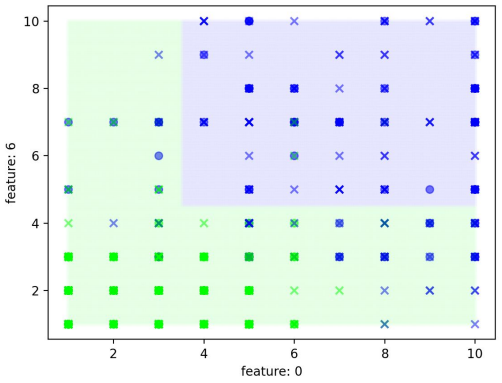


Figure 5: A Decision Boundary of Decision Tree Classifier. Feature 6 stands for bland chromatin; feature 0 represents clump thickness. Circles are training data and crosses are testing data. Instances in the blue area will be classified as malignant, whereas the green area is for benign ones.

From the two figures above, we can conclude that the decision boundaries of the KNN classifier are likely to be nonlinear and usually non-smooth. In contrast, the decision boundaries of the decision tree classifier are precisely linear, and the regions bounded by its decision boundaries are exactly hyper-cuboids.

3.5 Additional Experiments

This section includes additional experiments conducted on both models.

3.5.1 Scaling and Voting

Scaling is essential since KNN classifiers are extremely sensitive to the scaling of every feature. Therefore, we normalized the values of all features to reduce inductive bias.

Besides, for KNN classifiers, voting methods become critical when the test data point falls near to the decision boundary. Instead of the default uniform voting method, where K nearest neighbors have equal weights in classifying the test data, we developed the inverse distance weighting method, so among k

nearest neighbors, those closer to the test data are weighted more since their weights are proportional to the inverse of distance between them and the test data.

Accuracy(%)	Uniform	Inverse Distance
Breast Cancer	**94.45±1.62%**	**95.32±1.37%**
Hepatitis	**85.55±6.29%**	**86.48±6.33%**

The result above shows inverse distance voting outperforms over uniform voting.

3.5.2 KD Tree

By deploying KD tree, for a fixed dimension D, the time complexity of the searching process in KNN classifiers was reduced to $O(K \cdot \log N)$ from $O(KN)$ if the data has a balanced distribution. It greatly boosts the efficiency of the KNN classifier for large datasets.

3.5.3 Other Pre/Post-Pruning Approaches

This part we tried other pre/post-pruning approaches. One method is to set a minimum number of data in each leaf (N) for the decision tree and stop splitting when the data in the leaf is less than N.

This experiment tested the effect of the minimum number of data in each leaf (N) on the accuracy, and the maximum depth had been set to 100 in order to minimize its influence. The cost function is misclassification rate. Here we used the breast cancer dataset.

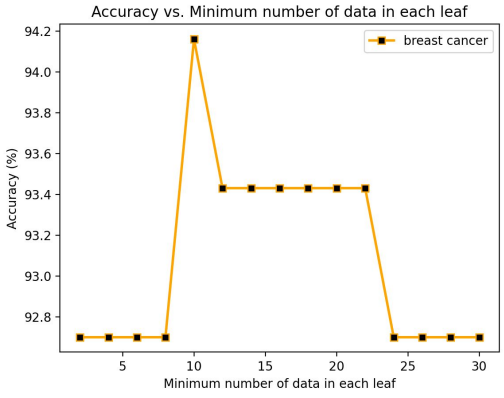
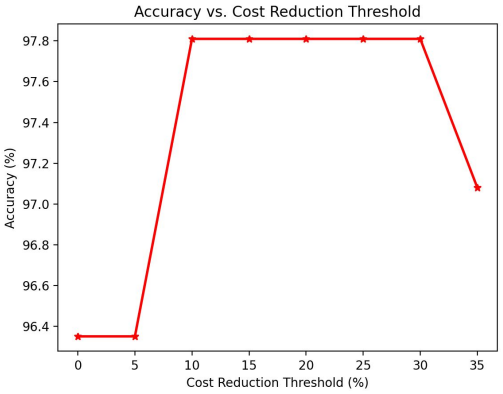


Figure 3: Accuracy vs. Minimum Number of Data for Each Leaf

This graph illustrates that the model hits the optimal accuracy for N=10. When N<10, the model overfits since smaller N allows more splits and leads to a deeper decision tree. As N increases, the accuracy drops due to underfitting because larger N makes splitting more difficult, leading to a shallow tree.

Also, another pre-pruning approach is to stop splitting if the reduction in cost is small (e.g. the reduction is less than 20%). The following experiment uses the breast cancer dataset. This approach works similar to minimum leaves.



Besides, another post-pruning strategy is greedily merging the leaves of the trained decision tree with their parents until it hurts the accuracy of the model on the validation set. However, this approach does not seem to work very well on both datasets. A comparative analysis of pruning methods conducted by Floriana, Donato, and Giovann verified this result, because both datasets are pruning insensible.

4. Discussion and Conclusion

To summarize, KNN and decision trees reached similar accuracy on some datasets like breast cancer, but KNN is slightly better in classifying datasets like hepatitis. For KNN, only an intermediate K value gives the optimal accuracy; for decision trees, small maximum depth underfits and large maximum depth leads to overfitting, and finally loses its effects. Different cost/distance functions will impact accuracy and sometimes simplicity wins. The decision boundary of KNN classifiers is often nonlinear and non-smooth, but that of decision trees is linear and orthogonal to the axis of the feature used for splitting. KNN classifier is very sensitive to scaling, so normalization will reduce inductive bias and improve its performance. Optimization of KNN can also be achieved by using efficient data structure and good voting strategies. For decision trees, other ways of pruning can also improve the decision tree and some of them have effects similar with maximum depth, whereas others have limited effects depending on the datasets.

5. Statement of Contribution

Pingsheng Li: implementation and optimization of decision tree models, and experiments related to the decision tree model.
Xiyuan Feng: cross-validation of the decision tree, analysis and summary of experimental data, compilation of the report.
Jacob Wang: implementation and optimization of KNN models, experiments design.

References

Floriana Esposito and Donato Malerba and Giovanni Semeraro. *A Comparative Analysis of Methods for Pruning Decision Trees*. IEEE Trans. Pattern Anal. Mach. Intell, 19. 1997.

David B. Skalak. *Prototype Selection for Composite Nearest Neighbor Classifiers*. Department of Computer Science University of Massachusetts. 1997.