Implementation and Comparison of KNN and its Improved Algorithm

Xiaonan Hu, Liqi Zhu

2017 Fall

1. Introduction

1.1 Background

As one of the Top 10 Algorithms in Data Mining, k-nearest neighbor (kNN) classification Algorithm has its almost simplest and comparatively sophisticated theory, so that it is a supervised learning classification algorithm can always be easily realized.

The basic theory of kNN classification algorithm is to find a group of k objects in the training data set that are closet to the test object, and base the assignment of a label on the predominance of a class in this neighborhood. So, there are three elements need to be fulfilled to implement the algorithm, including a set of labeled data, a measurement of distance (similarity), the parameter k indicates the number of nearest neighbors.

When given an unlabeled object, using kNN to classify, we should firstly calculate the distance of the certain point to each labeled data, then sort the distances and identify the k-nearest neighbors, finally votes of the class labels determines the class of the required object.

Current existing research relating to the improvement of kNN classification algorithm can be concluded into five basic aspects as follow: First, simplify calculating methods or reduce computation load to improve efficiency of the algorithm; Second, define new measurement standard for distance or similarity; Third, modify the threshold definition of k-nearest neighbors; Forth, using advanced algorithm to select better value of parameter k; Fifth, combine with another algorithm.

1.2 Problem Statement

Classic KNN algorithm has its obvious drawbacks. It's a lazy leaner algorithm, which means it will not generate a fixed model to classify the data, resulting in a larger cost for computing and storage.

And, the traditional kNN algorithm use insufficient part of the information in training data leading to situations that data cannot be classified. Therefore, both in the aspect of efficiency and accuracy, kNN algorithm need to be improved.

For this project, we will implement one improved algorithm of kNN classification based on information entropy of attributes values, which integrates developments target at two of the potential aspects mentioned before: similarity measurement and neighbor discrimination, and try to provide a feasible idea for the further improvements of kNN.

2 Related Works

KNN algorithm has the advantages of simpleness and easiness to understand and implement principle, not necessary to make assumptions or parameters estimation of training data, has relatively high precision, especially suitable for multi-classification problems.

However, the KNN algorithm belongs to the negative learning algorithm, so it needs a large amount of computation in the test sample classification process, the corresponding memory overhead is relatively large, the corresponding operation process is slow, and the application rate of the data information is low, so the accuracy needs to be improved, and the interpretability of the classification results is poor. Aiming at the shortcomings of KNN algorithm, the early research mainly improves kNN algorithm from the following aspects:

1) Reduce the computational overhead to improve the efficiency of the algorithm

KNN algorithm belongs to lazy learning classification algorithm, it needs to store all training set sample point data in the process of algorithm implementation, and needs to calculate the distance based on the training sample set for each sample to be classified, so the traditional KNN algorithm will cause huge storage occupation and computing overhead.

In response to this problem, the improved KNN algorithm proposed in the existing research can generally be divided by two types of improvement directions. One is to reduce the size of the training set. Usually, such methods include: delete a part of the samples with less relevance to the classification from the original training sample set; select part of the representative samples from the training set to form a new training sample set, then the clustering method is used to process the original training sample set to generate the cluster center points as a new set of training samples. The improvement of the cutting method of training samples based on density-based KNN text classifier proposed by Zhongyang Xiong et al. is such a method, which improves the cutting strategy based on the method of Ronglu Li et al. The sample area is supplemented accordingly, significantly improving the accuracy and stability of the final KNN classification.

The other is the algorithm for fast searching K nearest neighbors, these methods include: the method of partial distance calculation, such as WJ Hwang proposed. By performing a partial distance search in the wavelet domain for k nearest neighbors, the proposed method effectively improves the computation time down to 12.94% of the original time. We can also apply KNN improved algorithm that introduces an efficient indexing method. This kind of method can greatly reduce the computation cost of K neighbors and is capable of process huge training data set, especially in the case of high-dimensional space. And based on the inverted grid index and the MapReduce-based spatial KNN algorithm in the cloud environment proposed by Biao Liu, the establishment of the index in this method is significantly faster than R-tree and Thiessen polygon index establishment with better scalability; based on this KNN algorithm, the time to process the spatial query process is significantly reduced. It's easy to aware that the main advantage of this method is that it can reduce the computational overhead and improve the classification algorithm's running speed. However, the disadvantage of this kind of method is that such calculation or search can not guarantee the result is the global optimum.

2) Optimize the similarity measure method

The traditional KNN algorithm usually adopts the Euclidean distance in the process of classification. This similarity measure method causes the traditional KNN algorithm to be relatively sensitive to noise characteristics. For the similarity measure of Hamming distance commonly used for discrete variables, it is defined as an integer so that the value of its relative difference in the range of values is relatively small, it is more likely to appear as the same distance to affect the voting results, resulting in the phenomenon of error categories with most final votes. In addition, there shouldn't be a big difference in the range of each variable in the training sample point where the impact on distance will be correspondingly increased, leading to the final classification result. When this happens, we can simply eliminate the effects by standardizing the data.

To overcome the defects of similarity measure of traditional KNN algorithm, we usually take the method of giving the corresponding weights to different features. The determination of feature weight is usually based on the decision-making function of the classification. The weight can be set according to the entire training sample set or the local training sample set of the nearest neighbor of the sample to be tested. The existing researches studied a variety of methods for learning weight adjustment.

3) Optimize the decision strategy

The traditional KNN algorithm considers only the number of training samples of each class in the K nearest neighbors in the decision-making process. This kind of decision-making method is relatively easy to cause the deviation of the decision-making. The obvious flaw in this way is that simple counting can easily lead to the same number of votes being cast in different categories, making the result difficult to determine. This problem is the easiest to solve considering that the closer the sample point is, more likely it is to be classified as the sample class as the sample point to be classified. The method of weighting the vote by distance is often adopted. That is, the closer the distance is, the higher the weight is. This kind of weighted voting may improve the accuracy of the classification results.

However, in practical applications, the training sample concentrated data often have the phenomenon that the number of each type of data is unbalanced, which may result in that the KNN may be easier to obtain the votes of categories with more training samples in the decision-making process, thus causing the sample points Misclassification. Secondly, even if we obtain a set of training samples that are approximately equal in the number of different types of samples, the density of samples in each sample still lags and the accuracy of the algorithm will decline. At present, the solution to the problem of inhomogeneous training samples is to homogenize the distribution density of samples. In addition, the algorithm of iterative refinement for KNN coefficients proposed by Chaoyang is that the misclassified samples should be closer to the corresponding class and their 'distance' between the misjudged class should be increased, to improve the classification decision of the sample to be tested, and to correct the influence of the density of the training sample set.

4) Select the appropriate K value

K in KNN algorithm is a self-defined constant whose value will directly affect the final classification result. When the K value is too small, the classification result is more likely to be affected by noise, resulting in a decrease in the classification accuracy; while when the k value is too large, the neighbor may contain too many other types of points to increase the noise, Resulting in reduced classification effect.

In the practical application, a more scientific and rigorous method to determine the value of K is the use of cross-validation error statistical selection method: This method is that training data samples are divided into training samples and test samples as two parts at first, then get a machine learning model with the new training set of samples, and then use the corresponding set of test samples to test and calculate the error rate, change K for different values for the test, K value is the most appropriate data under the current conditions with the smallest error rate.

5) A variety of algorithms integrated

The current research on the improvement of KNN algorithm is not limited to optimizing the KNN algorithm from all aspects, many other improved algorithms try to combine other algorithms with KNN algorithm to improve accuracy and efficiency, which includes the combination of SVM and KNN algorithm, integrating KNN with Grouping Latent Semantic Analysis, integrating genetic algorithm with fuzzy KNN, and combining classification algorithm of decision tree and KNN.

3. Methodology

3.1 Entropy-KNN algorithm concept introduction

According to the definition of information entropy, it can be used to measure the importance of attribute value to class decision. The smaller the entropy is, the stronger the decision-making ability of corresponding attribute value is. The Entropy-KNN algorithm based on information entropy is used to redefine the measure of similarity, replacing the traditional KNN algorithm with the average information entropy of the same attribute value shared by any two samples. Entropy-KNN algorithm selects the K training samples nearest to the sample to be tested according to the distance defined based on the entropy of the eigenvalue. The average distance between neighbor sample and sample to be classified and the portion of frequency are considered complementarily to determine the class of sample to be tested.

The algorithm is defined as follows:

Definition of attribute information entropy: For dataset S, there are n classes $C_1, C_2, \ldots, C_n$, let attribute V have i different values $\{v_1, v_2, \ldots, v_n\}$, frequency of v_i is recorded as $|v_i|$, where the number of values belong to C_j is denoted as $|v_{ij}|$, then the information entropy of attribute v_i is:

$$S(v_i) = -\sum_{j=1}^{n} p_{ij} \ln(p_{ij}) \quad (1)$$

Where $p_{ij} = \frac{|v_{ij}|}{|v_i|}$, that is the probability that the sample whose attribute V value is v_i belongs to $C_j$class. When $|v_i| = |v_{ij}|, S(v_i) = 0$.

According to the definition of attribute value information entropy, if the samples with specific attribute value all belong to the same class, the information entropy of this attribute value is zero, which means that the data sample possessing this attribute value can be completely determined as the corresponding class ; When the attribute value information entropy is smaller, more probable this attribute value decides to classify the sample points into a specific class.

Based on the distance defined by the attribute information entropy: Let A, B be any two samples. The same attribute values of $A\ and\ B$ are $x_1, x_2, \dots, x_m$. The distance between A and B is defined as follows:

$$D_{Entropy}(A, B) = \frac{1}{m}\sum_{i=1}^{m} S\,(x_i) \tag{2}$$

The distance between any two sample points is defined as the average information entropy of all the same attribute values.

The distance defined by the information entropy based on the attribute value effectively measures the similarity between the two samples by the same attribute value between the two samples and the decisive effect of the attribute value on the classification. The closer the distance between A and B, the smaller the average information entropy of the same attribute value of the two sample points is, which means that there are more same attribute values with higher decisiveness within two samples. Therefore, it is more likely that A and B belong to the same class.

Class credibility: Let C_j be the class, B be the sample to be tested, A_i be the samples belonging to the C_j class in the neighbor samples, K be the total number of the nearest neighbor samples, and K_j be the number of samples belonging to the C_j class in the neighbor samples. Let $T(C_j, B)$ be the credibility of B against C_j, and the formula is as follows:

$$T(C_j, B) = \frac{K-K_j}{K} \times \frac{1}{K_j} \sum_{i=1}^{K_j} D_{Entropy}(A_i, B) \quad (3)$$

Where $\frac{1}{K_j} \sum_{i=1}^{K_j} D_{Entropy}(A_i, B)$ is the average distance between B and class C_j neighbor samples.

According to the definition of class credibility, calculation of $T(C_j, B)$ is composed of two parts, first part is the proportion of $\text{non}C_j$ samples in the neighbor samples, the smaller the value of this part is, the more C_j samples are in the neighbor samples , B is more likely to be classified as this class; the second part is the average distance between the sample under test and C_j samples, the smaller the value, the greater the probability that B belongs to C_j class. To sum up, the confidence level of the class defined in Eq. (3) is a class discriminant that combines the number of C_j samples and the average distance information of B and C_j samples. The smaller the $T(C_j, B)$ is, more likely it is that B belongs to the C_j class.

Entropy-KNN algorithm firstly applies formula (1) to calculate the training sample set, and obtains all attribute value information entropy, and then compares the sample point to be measured with the training set to obtain the information of the same attribute value between points and points, then use formula (2) to calculate the distances $D_{Entropy}(A, B)$ of each sample from the test sample points and the training set respectively, arrange all the obtained distances in ascending order and select the first K nearest neighbor training samples, calculate the number of each kind of neighbor samples K_j. Finally use the formula (3) to calculate the confidence level $T(C_j, B)$ of the sample to be tested and each class, and the class corresponding to the minimum value of $T(C_j, B)$ is B's classification result.

3.2 Entropy-KNN Algorithm Analysis

The Entropy-KNN algorithm based on attribute value information entropy improves the distance calculation method based on information entropy, and defines the concept of class credibility as classification criteria.

The definition of distance in traditional KNN algorithms, whether Euclidean distance or Hamming distance, only a certain point in the training sample and sample to be tested was considered of being tested (I.e. only the point-to-point similarity); however, the attribute information entropy can effectively measure the importance of attribute value to the classification decision. Distance measure defined by Entropy-KNN algorithm based on the attribute value of information entropy is equivalent to indirect extraction of all the information with specific attribute values of training samples to provide a reference for classification for the sample point. Compared with the traditional KNN algorithm, the Entropy-KNN algorithm makes full use of the feature information of the training sample set and quantifies the corresponding weights of the eigenvalues for the classification decision.

In addition, the Entropy-KNN algorithm is also optimized in the classification decision strategy. Traditional KNN algorithm uses simple voting counting method and distance-weighted KNN algorithm simply considering the distance voting method, Entropy-KNN combines the decision-making criteria of the first two algorithms simultaneously, considering number of trainging samples of each class in K Neighbors and the average distance between the class to be classified and each class of K nearest neighbor. This definition of decision index can effectively avoid the influence of noise point in K neighbors on decision. It's less likely that with the same decision index, the class can not be determined.

In summary, KNN algorithm based on improved information entropy can theoretically improve the accuracy and discriminant efficiency of KNN algorithm. In addition, it can be easily seen from the definition of Entropy-KNN algorithm that Entropy-KNN algorithm is only suitable for data set with discrete or qualitative variables with finite values. This algorithm may not be suitable for the case of too many classes, which may result in multiple classes including specific attribute values, leading to relative reduction or the similarity of the attribute value's importance to each class. The accuracy of the final determination result may be further interfered.

4 Experiment

4.1 Data Sets Description

This project will do research based on two data sets for classification, and with discrete variables for each attribute.

The first data set is the Letter Recognition Data Set from UCI, which contains 20000 samples, each of which characters image features classified by 26 letters, 16 attributes in total. And all the value of attributes take the integers range in 0-15. Each attribute's name and meaning shows in following table:

Table 1: Attributes of the Letter Recognition Data Set

Attribute	Description
Y	letter：capital letter (26 values from A to Z)
X_1	x-box：horizontal position of box (integer)
X_2	y-box：vertical position of box (integer)
X_3	width：width of box (integer)
X_4	high：height of box (integer)
X_5	onpix：total # on pixels (integer)
X_6	x-bar：mean x of on pixels in box (integer)
X_7	y-bar：mean y of on pixels in box (integer)
X_8	x2bar：mean x variance (integer)
X_9	y2bar：mean y variance (integer)
X_{10}	xybar：mean x y correlation (integer)
X_{11}	x2ybr：mean of x*x*y (integer)
X_{12}	xy2br：mean of x*y*y (integer)
X_{13}	x-ege：mean edge count left to right (integer)
X_{14}	xegvy：correlation of x-ege with y (integer)
X_{15}	y-ege：mean edge count bottom to top (integer)
X_{16}	yegvx：correlation of y-ege with x Y(integer)

The second data set is for Predicting 5-Year Career Longevity for NBA Rookies, which contains 1340 samples, to facilitate the following experiments, we only take 11 attributes and level them into 10 level as integers range in 0-9. And the data will fall in two classes as whether a NBA player will last 5 years in league. All the attributes take and their meanings shown in following table:

Table 2: Attributes of the NBA Player Data Set

Attribute	Description
TARGET_5Yrs	1 if career length >=5, 0 if career length <5
GP	Games played
MIN	Average minutes played
PTS	Average points per game
FG%	Field goal percentage
3P%	3-point goal percentage
FT%	Free throw goal percentage
REB	Average rebounds per game
AST	Average assists per game
STL	Average steals per game
BLK	Average blocks per game
TOV	Average turnovers per game

4.2 Experiment Design

As the algorithm always tend to assign the target point into the class with higher number of objects, the metrics of data set distribution should be checked as the first step, and every experiment should be implement on relative balanced data sets, to reduce the unwilling influence. For both kNN and distance-weighted kNN in this experiment, Euclidean distance will be taken as the distance measurement, and simple vote for the decision process.

Generally, the experiments for this project will be divided into three major portions. The first part of the experiment, we will set a relatively large size of training data, and then compare the accuracies of classifiers: kNN, distance-weighted kNN and Entropy kNN, based on the same test data set. The second part of experiments will be respectively implement the three algorithms on different size of training data, based on the accuracies acquired, the influence of training data sets' size for each algorithm can be observed. The last part of experiments, targets on the influence of value k, will reveal the effect of k value for each algorithm, based on relatively smaller training data sets. The former three portions of experiments are all implement on the letter recognition data set, and the fourth part experiment will be applied in the other NBA data set, so that we can acquire a view of how data sets' characteristics will affect the classification algorithm's accuracy.

To acquire relatively objective results for the experiments, 10-folds cross validation will be implemented. The whole data set will be petitioned into ten folds, every experiment will take one of the folds as training data set, and the others as test data set. The average accuracy for ten times experiments will be recorded and compared.

4.3 Experiment Results

The first part experiment take 10 sets of data, each contains a training data set of 2000 and a testing data set of 18000. Implement all three algorithms with k=10, and record the 10 accuracies for each algorithm, calculate the average and compare.

Table 3: Accuracies for k=10, training data size=2000

No./Acuracy	kNN	Distance-weighted	Entropy-kNN
1	0.737	0.778	0.842
2	0.743	0.784	0.817
3	0.754	0.796	0.817
4	0.760	0.799	0.805
5	0.767	0.812	0.828
6	0.752	0.790	0.783
7	0.755	0.797	0.801
8	0.761	0.795	0.826
9	0.774	0.812	0.782
10	0.761	0.804	0.792
Average accuracy	0.756	0.797	0.809

As the results shown in Table 3, when k=10 and training data size=200, compared to classic kNN algorithm, distance-weighted kNN improves the performance in accuracy, 4.1 percent in average; however, Entropy-kNN elevated less compared to distance-weighted kNN algorithm. So that we can come to a conclusion that, In the second part of the experiment, we randomly select 10 datasets with sample sizes of 1000, 2000, ..., 10000 from the dataset and then perform the same steps as the first part on each dataset, so we can get the classification accuracies of KNN algorithm, distance-weighted KNN algorithm and Entropy-KNN algorithm with the training data set size of 100, 200, ..., 1000 for comparative analysis.

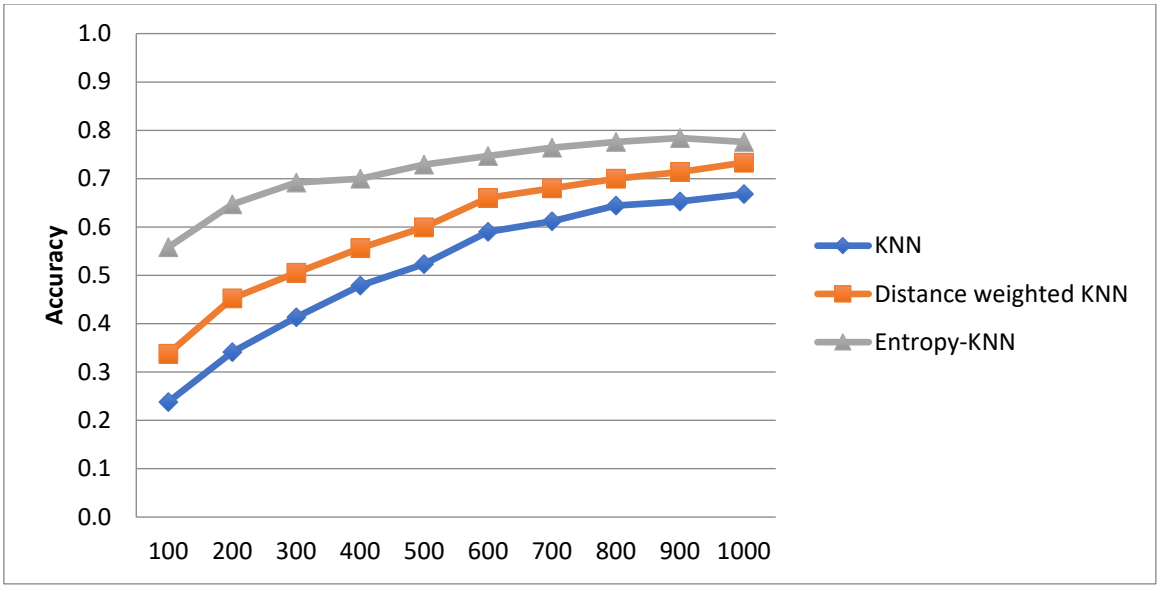

Figure1: k=10, Classification accuracies on different sizes of training data

The second part of the experimental comparison of the accuracy of the three algorithms shown in Figure 1. Comparing the three average accuracy curves of KNN, distance-weighted KNN and Entropy-KNN, it is easy to see that the Entropy-KNN algorithm has more accuracy advantages when the training set is smaller. Thus, compared with the traditional KNN algorithm, the Entropy-KNN algorithm can dig and use more information mining from the training set information, and still can extract sufficient available information in a small training data set, to effectively improve the accuracy of the classification algorithm. Therefore, the Entropy-KNN algorithm is a more suitable classification algorithm when training data samples are relatively scarce.

In the third part of experiment, the method is as follows: Firstly, randomly select 2000 samples from the dataset as the new dataset, and then perform 10-fold cross validation on the new dataset. Run kNN algorithm, distance-weighted kNN algorithm and Entropy-kNN algorithm, and take k value respectively as 5,10, ..., 50. Finally, we can get the experiment results within a training set size of 200, using different values for parameter k. Then we do the comparative analysis of accuracy for kNN, distance-weighted kNN and Entropy-kNN respectively.

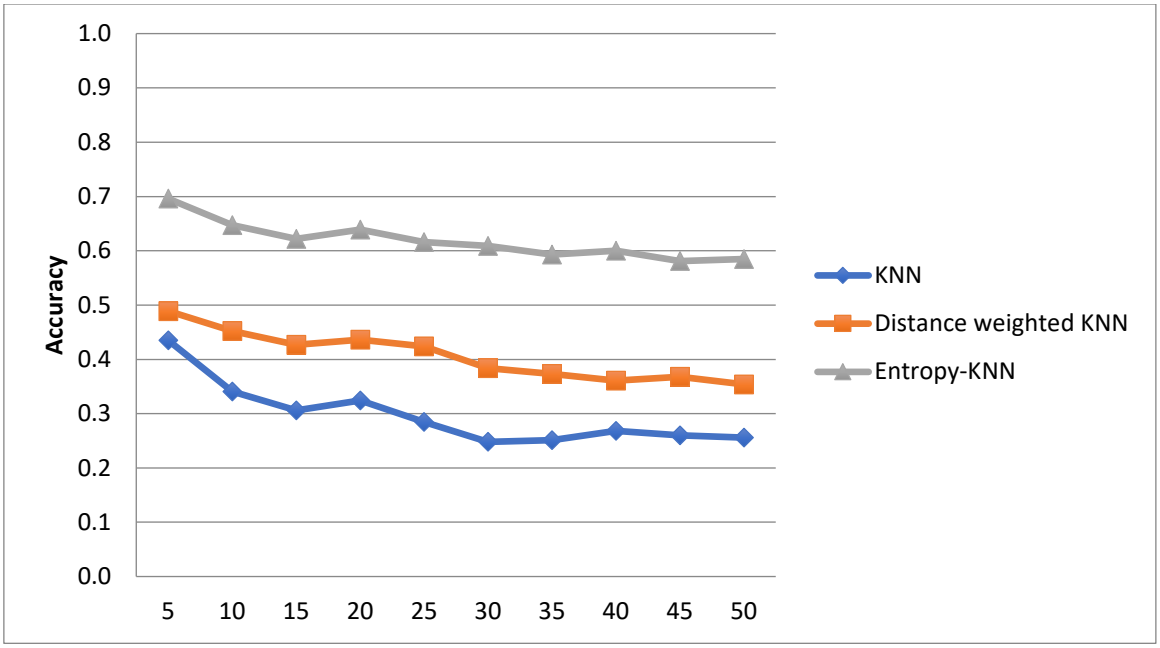

Figure 2: Training data size=200, Classification accuracies with different k

As experiment results shown in Figure 2, it can be easily seen that when the training data sets are small, both the traditional KNN algorithm and the distance-weighted KNN algorithm are more likely to be affected by the K value; that is, when the K value increases and there are too many points belonging to other classes are included in the neighborhood. These two method is more susceptible to noise and results in a noticeable decreased accuracy in classification mainly because that kNN algorithm and distance-weighted KNN algorithm only use simple voting method for classification decision, by contrast, the decision-making method used by Entropy-KNN algorithm introduced the definition of class credibility, which can make fully use of the information from smaller training data set, and also excluding the interference of noise points included when taking too many neighbors.

The last part of experiment implemented on the NBA data set, with training data size of 134 and k=10, still using kNN, distance-weighted kNN and Entropy-kNN respectively, and evaluate with 10-fold cross validation. The experiment results shown as follow:

Table 4: Accuracies for k=10, training data size=134

No./Acuracy	kNN	Distance-weighted	Entropy-kNN
1	0.645	0.710	0.958
2	0.651	0.716	0.933
3	0.662	0.728	0.933
4	0.668	0.732	0.921
5	0.675	0.743	0.944
6	0.660	0.722	0.899
7	0.663	0.729	0.917
8	0.669	0.727	0.942
9	0.682	0.744	0.898
10	0.669	0.737	0.908
Average accuracy	0.664	0.729	0.925

From this part of experiment, we achieved better result and relatively large improvement with Entropy-kNN compared to kNN and distance-weighted kNN. After analysis, we conclude the main reason is that the dataset used in this experiment has less data classes with sufficient attributes, however, the letter recognition data set has relatively more classes, which leads to a generally increase on attributes information entropy and so to the relative dispersed decision power.

5. Conclusion

The Entropy-KNN algorithm is optimized for both the distance measurement method and decision-making method in the traditional KNN algorithm. Firstly, Entropy-KNN algorithm extracts the information of the importance of each attribute value to the class decision based on the training sample set, so as to define the similarity between the sample under test and the training sample to be measured by the average information entropy of the same attribute value therebetween. We use the training sample set information more efficiently to extract the nearest neighbor sample which is similar to the sample to be tested. Secondly, we define a new class decision criterion - class credibility considering the number and average distance of neighbor samples of each class.

Therefore, it is more effective and accurate to distinguish the credibility of the samples to be tested and to further improve the accuracy of the classification.

Comparing the above three experiments on UCI's letter recognition dataset, it can be known that Entropy-KNN has obvious improvement in classification accuracy compared with traditional KNN algorithm, especially when the existing training sample information is limited (ie. when the training sample set is small), Entropy-KNN algorithm outperforms the traditional KNN algorithm and distance-weighted KNN algorithm obviously. When the size of training sample set increases gradually, The relative advantage of Entropy- KNN algorithm is gradually reduced. Therefore, with a large training sample set, the KNN algorithm or the distance-weighted KNN algorithm can be selected within the allowable error range. In addition, the Entropy-KNN algorithm has a relatively small sensitivity to the selection of the K value, and the Entropy-KNN algorithm can still obtain a higher classification accuracy when the K value increases, while the KNN algorithm and the distance-weighted KNN algorithm are more likely to be interfered by increased noise sample point.

However, the Entropy-KNN algorithm performs more cyclic during calculation, resulting in less efficiency. The comparison between the experimental data, by comparing the results of letter recognition data set and NBA data set, we found that the improvement effect of Entropy-KNN in the experiments on the former one data set is relatively insignificant. After analysis, the main reason is that Data sets have more data types and eigenvalues, which leads to the relative dispersion of attribute values for each class and information entropy of attribute values is generally increased. Based on the above two issues, I think one of the most feasible ways is to pre-process the training sample set accordingly, such as clustering training sample sets. Such processing can not only reduce the computational complexity to effectively improve the efficiency of the algorithm, but also effectively solve the problem that the Entropy-KNN algorithm has a poor performance in the case of a large number of eigenvalues.

Although the traditional KNN algorithm has a simple and easy-to-understand theoretical basis, it has the disadvantages of complex computation, low efficiency and

low accuracy. In this paper, Entropy-KNN, an improved KNN algorithm based on the concept of information entropy, is selected for our experimental analysis. Then a total of four experiments were carried out on UCI's letter recognition dataset. The results show that the Entropy-KNN algorithm can effectively improve the classification accuracy of KNN algorithm. Especially when the training sample set is small, its performance is obviously better than the traditional KNN algorithm and distance-weighted KNN algorithm; Entropy-KNN algorithm has a relatively small sensitivity to the selection of K value, Entropy-KNN algorithm can still obtain higher classification accuracy when K value increases. However, the Entropy-KNN algorithm still needs to be further improved in terms of computational efficiency, and the analysis suggests that the clustering of training sample sets is a possible improvement of the algorithm.

6. Bibliography

[1] Wu X,Kumar V,Ross Quinlan J,et al.Top 10 algorithms in data mining[J]. Knowledge and Information Systems,2008,14(1).
[2] Wei L., Zhao X., Zhou X. (2015) An Enhanced Entropy-K-Nearest Neighbor Algorithm Based on Attribute Reduction. In: Wong W. (eds) Proceedings of the 4th International Conference on Computer Engineering and Networks. Lecture Notes in Electrical Engineering, vol 355. Springer, Cham
[3] k-Nearest Neighbour Classifiers –Pádraig Cunningham, Sarah Jane Delany — 2007
[4] Application of k-Nearest Neighbor on Feature –Projections Classifier To, Tuba Yavuz, H. Altay Guvenir — 1998 — Proceedings of ISCIS-98, 13th International Symposium on Computer and Information Sciences
[5] MKNN: Modified K-Nearest Neighbor –Hamid Parvin, Hosein Alizadeh, Behrouz Minaei-bidgoli
[6] k Nearest Neighbor Classification across –Multiple Private Databases, Li Xiong — 2006 — In Proceedings of ACM International Conference on Information and Knowledge Management
[7] Classification with Learning k-Nearest Neighbors –Jorma Laaksonen, Erkki Oja — 1996
[8] Extending the K-Nearest Neighbour Classification –Algorithm To Symbolic, Floriana Esposito, Donato Malerba, Marianna Monopoli.