*K*NN-CF Approach: Incorporating Certainty Factor to *k*NN Classification

Shichao Zhang

***Abstract*—*K*NN classification finds *k* nearest neighbors of a query in training data and then predicts the class of the query as the most frequent one occurring in the neighbors. This is a typical method based on the majority rule. Although majority-rule based methods have widely and successfully been used in real applications, they can be unsuitable to the learning setting of skewed class distribution. This paper incorporates certainty factor (CF) measure to *k*NN classification, called *k*NN-CF classification, so as to deal with the above issue. This CF-measure based strategy can be applied on the top of a *k*NN classification algorithm (or a hot-deck method) to meet the need of imbalanced learning. This leads to that an existing *k*NN classification algorithm can easily be extended to the setting of skewed class distribution. Some experiments are conducted for evaluating the efficiency, and demonstrate that the *k*NN-CF classification outperforms standard *k*NN classification in accuracy**.

***Index Terms*—Classification, *k*NN classification, imbalanced classification.**

I. Introduction

GIVEN its simplicity, easy-understanding and relatively high accuracy, the k-nearest neighbor (*k*NN) approach has successfully been used in diverse data analysis applications [4,10,31,35] such as information retrieval, database, pattern recognition, data mining and machine learning. In information retrieval application proposal, the *k*NN approach is used to, for instance, similarity searching [42], text categorization, ranking and indexing [2,61]. In database application proposal, the *k*NN approach is used to, such as, approximate query and high dimensional data query [11,49]. In pattern recognition application proposal, the *k*NN approach is used to, for example, segmentation and prediction [13,45]. In data mining and machine learning application proposal, the *k*NN approach is used to, for example, clustering and classification [14,22,23,26], manifold learning [50,57,58], and missing data imputation for data preparation [64,65]. Therefore, it has recently been recognized as one of top 10 algorithms in data mining [60].

The *k*NN method provides k data points in a given dataset most relevant to a query in a data analysis application. Without other information, the k most relevant data are k nearest neighbors of the query in the dataset. And then predicts the class of the query as the most frequent one occurring in the neighbors. This is a typical method based on the majority rule. Majority-rule based methods have widely and successfully been used in real applications. They can, however, be unsuitable to the learning setting of skewed class distribution. This is illustrated with Example 1 as follows.

Eample 1. Consider some data drawn from a dataset with skewed class distribution, as listed in Table I, or charted in Fig. 1. In Table I, X1 and X2 are two attributes, C is the class attribute (or decision attribute), "+" and "–" stand for the two classes, "?" denotes the unlabeled class.

TABLE I

Data from the questionnaire survey

X1	3	4	4	4	6	7	7	8	4	5	4
X2	7	3	4	10	2	4	9	5	6	5	5
C	+	+	+	+	+	+	+	+	–	–	?

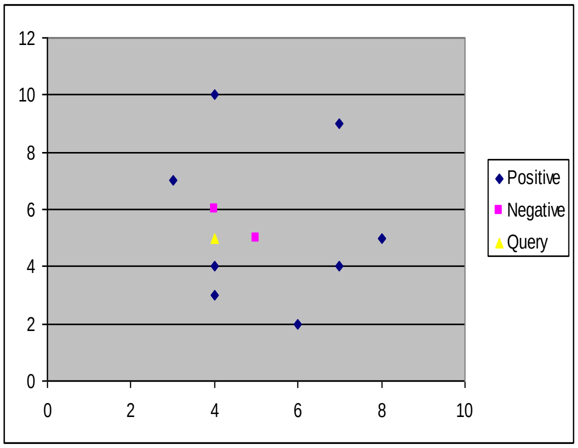

Fig. 1. Plotting the data in Table

Assume k = 5. For the query (4, 5, ?), we can obtain its 5 nearest neighbors in Table I, (3, 7, +), (4, 3, +), (4, 4, +), (4, 6, –), (5, 5, –). According to the *k*NN classification, "+" is the most frequent one occurring in the neighbors. Consequently, the class of the query (4, 5, ?) is predicted as "+". The first

Shichao Zhang is with the College of Computer Science and Information Technology, Guangxi Normal University, PR China; the State Key Lab for Novel Software Technology, Nanjing University, PR China;
e-mail: zhangsc@mailbox.gxnu.edu.cn.

feedback seems to be that the class of (4, 5, ?) should be predicted as "–", although the majority rule predicts it as "+".

To attack the above actual and challenging issue, this paper incorporates certainty factor (CF) measure to *k*NN classification, denoted as *k*NN-CF classification. The main idea is as follows. We have p(C = +) = 0.8 and p(C = –) = 0.2 in the dataset in Table I. The selected 5 nearest neighbors can be taken as a new evidence, E, and p(C = +|E) = 0.3 and p(C = –|E) = 0.2. Clearly, compared with their prior probabilities, the conditional probability of "–" is lifted much more than that of "+". Accordingly, it is reasonable to predict "–" as the class of (4, 5, ?). The CF measure can capture this ad hoc nature. And we will be incorporated to the a *k*NN classification in this paper, called ***k*NN-CF classification**.

This *k*NN-CF strategy can be applied on the top of a *k*NN classification algorithm (or a hot-deck method, or an instance-based algorithm) to meet the need of imbalanced learning. This leads to that an existing *k*NN classification algorithm can easily be extended to the learning setting of skewed class distribution. Some experiments are conducted for evaluating the efficiency, and demonstrate that the *k*NN-CF classification outperforms standard *k*NN classification in accuracy.

The rest of this paper is organized as follows. Section II briefly recalls work mainly related to *k*NN classification, imbalanced classification and certainty factor measure. The *k*NN-CF classification is presented in Section III. We evaluate the *k*NN-CF classification with real datasets downloaded from UCI in Section IV. This paper is concluded in Section V.

II. Preliminary

This section presents some basic concepts and briefly recalls related work in *k*NN classification, imbalanced classification and certainty factor measure.

A. *Research into kNN Approach*

KNN approach has recently been recognized as one of top 10 algorithms in data mining [60], due to its high classification accuracy in problems with unknown and nonnormal distributions [16,26,31] and its wide applications [35,4]. While NN (nearest neighbor) classification suffers from the issue of overfitting, a more sophisticated approach, *k*NN classification [21], finds a group of k objects in the training set that are closest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighborhood [1,12]. There are three key elements of this approach: a set of labeled objects, e.g., a set of stored records, a distance or similarity metric to compute distance between objects, and the value of k, the number of nearest neighbors. To classify an unlabeled object, the distance of this object to the labeled objects is computed, its k-nearest neighbors are identified, and the class labels of these nearest neighbors are then used to determine the class label of the object [60].

The *k*NN classification has the remarkable property that under very mild conditions, the error rate of a *k*NN classifier tends to the Bayes optimal as the sample size tends to infinity. However, there are several key issues that affect the performance of *k*NN, mainly including the choice of k, predicting the class labels of new data, distance measure selection, and lazy learning. For details, please read the paper [60].

Therefore, many techniques have recently been developed for improving the *k*NN classification. Among them, distance measure selection is relatively hot research topic [16,24,27,36]. Currently a variety of measures such as Euclidean, Hamming, Minkowsky, Mahalanobis, Camberra, Chebychev, Quadratic, Correlation, Chi-square, hyperrectangle distance [41], Value Difference Metric [47], and Minimal Risk Metric [5] are available. However, no distance function is known to perform consistently well, even under some conditions [54]. This makes the use of *k*NN highly experience dependent. Various attempts have been made to remedy this situation. Among those, notably DANN carries out a local linear discriminant analysis to deform the distance metric based on say 50 nearest neighbors [24]. LFM-SVM also deforms the metric by feature weighting, where the weights are inferred from training an SVM on the entire data set [15]. H*k*NN applies the collection of 15-70 nearest neighbors from each class to span a linear subspace for that class, and then classification is done based not on distance to prototypes but on distance to the linear subspaces [53]. There are other kinds of distance defined by the property of data. Examples are tangent distance on the USPS zip code data set [44], shape context based distance on the MNIST digit data set [3], distances between histograms of textons on the CUReT data set [52], and geometric blur based distances on Caltech-101 [70]. Furthermore these measures can be extended by kernel techniques such as to estimate a curved local neighborhood [37], which can make the space around the samples further or closer to the query, depending on their class-conditional probability distributions. More recently, a new measure named neighborhood counting is proposed to define the similarity between two data points by using the number of neighborhoods [54]. Because features of high dimensional data is often correlated so that measure easily becomes meaningless, some approaches are designed to deal with this issue such as an approach that applies variable aggregation to define the measure [16,26]. Besides all kinds of measures above, the other strategy can be also applied to select nearest neighbors. For example, an approach is proposed that considers the geometrical placement of neighbors more than actual distances to appropriately characterize a sample by its neighborhood [43]. This approach is effective in some case, but it is conflict with our intuition when data is on manifold.

Other main improvements of *k*NN classification include, such as fuzzy set theory and evidential reasoning [68], measures for finding the better nearest neighbors [16,26,54], and local mean classifiers (LMC) [8,31,34,62].

B. *Research into Imbalanced Classification*

The class imbalance (or skewed class distribution) is relatively a new issue in data mining and machine learning. While it was recognized that the imbalance can cause suboptimal classification performance, there are many research reported on imbalanced learning since two workshops "AAAI'2000 Workshop on Learning from Imbalanced Data Sets" and "ICML'2003 Workshop on Learning from

Imbalanced Data Sets". The main efforts include, for example, the detection of software defects in large software systems [33], the identification of oil spills in satellite radar images [28], the detection of fraudulent calls [20], and the diagnoses of rare medical conditions such as the thyroid disease [40].

In the setting of skewed class distribution, it is obvious that the rare instances in these applications are of critical importance. And classification learning should be able to achieve accurate classification for the rare classes. Typically the rare instances are much harder to identify than the majority instances. Different from traditional classification desired a high overall accuracy, the purpose of imbalanced learning is to achieve accurate classification for the rare class without sacrificing the performance for other classes.

While existing classification algorithms work well on the majority classes, there have been several attempts to adjust the decision bias favourable to the minority class. Holte et al. [25] modified the bias of CN2 classifier, by using the maximum generality bias for large disjuncts and a selective specificity bias for small disjuncts. Another piece of work is by Ting [51], where a hybrid approach for addressing the imbalanced problem was proposed. This method adopted C4.5 as the base learner, then an instance-based classifier was used if small disjuncts were encountered. Similar approaches were proposed by [6,7], using a combination of the genetic algorithm and the C4.5 decision tree. However, their experimental results show no statistically significant difference from the base C4.5 learner.

Re-sampling techniques have been a popular strategy to tackle the imbalanced learning problem, including random over-sampling and under-sampling, as well as intelligent re-sampling. Chawla, et al. [9] proposed a synthetic minority over-sampling technique to over-sample the minority class. Kubat and Matwin [29] tried to under-sample the majority class. Another related work by Ling and Li [32] combined over-sampling of the minority class with undersampling of the majority class. However, no consistent conclusions have been drawn from these studies [55]. The effect of re-sampling techniques for active learning was analysed in [69]. They found over-sampling is a more appropriate choice than under-sampling which could cause negative effects on active learning. A bootstrap-based over-sampling approached was proposed, and it was shown to work better than ordinary over-sampling in active learning for word sense disambiguation.

The second strategy tackling the imbalanced distribution problem is cost-sensitive learning [17,19,66]. Domingos [18] proposed a re-costing method called MetaCost, which can be applied to general classifiers. The approach made error-based classifiers cost-sensitive. His experimental results showed that MetaCost reduced costs compared to cost-blind classifier using C4.5Rules as the baseline.

Ensemble learning has also been studied for imbalanced classification. Sun et al. [48] tried to use boosting technique for imbalanced learning, and three cost-sensitive boosting algorithms were introduced. These boosting algorithms were investigated with respect to their weighting strategies towards different types of samples. Their effectiveness in identifying rare cases on several real-world medical datasets with imbalanced class distribution were examined. An empirical study by Seiffert et al. [73] compared the performance between re-weighting and re-sampling boosting implementations in imbalanced datasets. They found that boosting by re-sampling outperforms boosting by re-weighting, which is often the default boosting implementation.

A potential strategy is the instance-based learning that will be built in Section III. The ubiquitous instance-based learning paradigm is rooted in the *k*NN algorithm. Most research efforts in this area have been on trying to improve the classification efficiency of *k*NN [1,59]. Various strategies have been proposed to avoid an exhaustive search of all training instances and to achieve accurate classification. However, to the best of our knowledge, no work has been reported adjusting the induction bias of *k*NN for imbalanced classification.

C. *Research into Certainty Factor Measure*

The certainty-factor (CF) model is a method for managing uncertainty in rule-based systems. Shortliffe and Buchanan [46] developed the CF model in the mid-1970s for MYCIN, an expert system for the diagnosis and treatment of meningitis and infections of the blood. Since then, the CF model has become the standard approach to uncertainty management in rule-based systems. A certainty factor is used to express how accurate, truthful, or reliable you judge a predicate to be. Note that a certainty factor is neither a probability nor a truth value. Therefore, it is slightly dodgy theoretically, but in practice this tends not to matter too much. This is mainly because the error in dealing with certainties tends to lie as much in the certainty factors attached to the rules (or in conditional probabilities assigned to things) as in how they are manipulated.

A certainty factor is a number between -1 and 1 (or in [-1, 1]) that represents the change in our belief on some hypothesis. A positive number means an increase in the belief and a negative number the contrary. A value of 0 means that there is no change in our belief on the hypothesis.

The CF measure has successfully been used to identify both positive and negative association rules in datasets [71,72]. This leads to the fact that a framework was built for complete association analysis (both positive and negative association rules).

III. KNN-CF Classification

For simplifying the description, we adopt the CF measure in [72] for building the framework of the kNN-CF classification. Before presenting the *k*NN-CF classification, a simple measure, called FR (frequency ratio), is introduced in Section III.A.

A. *KNN Classification Based on FR measure*

Let D be a training set, C = {c1, c2, …, cm} a set of labels, Q a query, N(Q, k) the set of k nearest neighbors, f(C= ci, D) the frequency of ci in D, and f(C= ci, N(Q,k)) the frequency of ci in N(Q, k). We define the FR measure as follows.

$$\text{FR}(C= c_i) = \frac{f(C=c_i, N(Q,k))}{f(C=c_i, D)} \quad . \qquad (1)$$

The FR strategy for *k*NN classification is defined as follows. We first obtain

$$S_{FR} = \{ \arg\max_{1 \le i \le m} \{FR(C = c_i)\} \}. \qquad (2)$$

Because there may be one more classes satisfy $\arg\max_{1 \le i \le m} \{FR(C = c_i)\}$, $|S_{FR}|$ can be greater than 1. Accordingly, we can predict the class c of Q with Formula (3) as follows.

$$c = \arg\max_{c_j \in S_{FR}} \{c_j\} \quad . \qquad (3)$$

We illustrate the use of FR measure with the data in Example 1. From Table I and the 5 nearest neighbors of the query, the frequency of classes "+" and "–" can be computed as follows:

f(C=+, D) = 8

f(C=–, D) = 2

f(C=+, N(Q,5)) = 3

f(C=–, N(Q,5)) = 2.

Consequently, we can obtain

FR(C= +) = $\dfrac{f(C=+, N(Q,5))}{f(C=+, D)} = \dfrac{3}{8}$ = 0.375

FR(C= –) = $\dfrac{f(C=-, N(Q,5))}{f(C=-, D)} = \dfrac{2}{2}$ = 1.

Therefore, we can predict the class c of the query Q as follows.

$$S_{FR} = \{ \arg\max_{1 \le i \le m} \{FR(C = c_i)\} \} = \{-\},$$
$$c = \arg\max_{c_j \in S_{FR}} \{c_j\} = -.$$

From the above, although class "+" is the most frequent one occurring in N(Q, 5), its frequency ratio, FR(C= +) = 0.375, is much low than that of class "–", FR(C= –) = 1. Therefore, it is reasonable to predict "–" as the class of (4, 5, ?).

The FR is a simple and efficient strategy. It is similar the "lift" measure in data mining and machine learning, which is a measure of the performance of a model at predicting or classifying cases, measuring against a random choice model (adopted from Wikipedia).

Certainly, we can replace FR with the odds ratio. The use of odds ratio to *k*NN classification is similar to that of FR strategy.

B. KNN Classification Based on CF measure

With the assumption in Section III.A, we incorporate the CF measure to *k*NN classification as follows. Assume p(C= ci |D) is the ratio of ci in training set D, p(C= ci |N(Q,k)) is the ratio of ci in the set of k nearest neighbors, N(Q, k). If p(C= ci |N(Q,k)) ≥ p(C= ci |D), the CF is computed with (4) as follows.

CF(C= ci, N(Q,k)) = $\dfrac{p(C = c_i \mid N(Q,k)) - p(C = c_i \mid D)}{1 - p(C = c_i \mid D)}$.

(4)

If p(C= ci |N(Q,k)) < p(C= ci |D), the CF is computed with (5) as follows.

CF(C= ci, N(Q,k)) = $\dfrac{p(C = c_i \mid N(Q,k)) - p(C = c_i \mid D)}{p(C = c_i \mid D)}$.

(5)

According to the explanation of CF, CF(C= ci, N(Q,k)) is valued in [-1, 1]. If CF(C= ci, N(Q,k)) > 0, our belief on that the class of the query should be predicted as C= ci is increased. CF(C= ci, N(Q,k)) < 0, our belief on that the class of the query should be predicted as C= ci is decreased. CF(C= ci, N(Q,k)) = 0, our belief on that the class of the query should be predicted as C= ci is the same as that in the training set D.

The CF strategy for kNN classification is defined as follows. We first obtain

$$S_{CF} = \{ \arg\max_{1 \le i \le m} \{CF(C = c_i, N(Q,k))\} \}.$$

(6)

Because there may be one more classes satisfy $\arg\max_{1 \le i \le m} \{CF(C = c_i, N(Q,k))\}$, $|S_{CF}|$ can be greater than 1. Accordingly, we can predict the class c of Q with Formula (7) as follows.

$$c = \arg\max_{c_j \in S_{CF}} \{c_j\} \; . \qquad (7)$$

Also, we illustrate the use of CF measure with the data in Example 1. Because f(C=+, D) = 8, f(C=–, |D) = 2, f(C=+, N(Q,5)) = 3 and f(C=–, N(Q,5)) = 2, we have p(C=+|D) = 0.8, p(C=–|D) = 0.2, p(C=+| N(Q,5)) = 0.6 and p(C=–| N(Q,5)) = 0.4. Because p(C=+| N(Q,5)) < p(C=+|D), we should calculate the CF of "+" with (3) as follows

CF(C= +, N(Q,5)) =

$\dfrac{p(C = + \mid N(Q,5)) - p(C = + \mid D)}{p(C = + \mid D)} = \dfrac{0.6 - 0.8}{0.8}$ = –0.25

Because p(C=–| N(Q,5)) > p(C=–|D), we should calculate the CF of "–" with (2) as follows

CF(C= –, N(Q,5)) =

$$\frac{p(C=-\mid N(Q,5))-p(C=-\mid D)}{p(C=-\mid D)}=\frac{0.4-0.2}{1-0.2}=0.25$$

Therefore, we can predict the class c of the query Q as follows.

$$S_{CF}=\{\ \underset{1\le i\le m}{\arg\max}\{CF(C=c_i,N(Q,k))\}\ \}=\{-\}$$
$$c=\underset{c_j\in S_{CF}}{\arg\max}\{c_j\}=-.$$

From the above, although class "+" is the most frequent one occurring in N(Q, 5), its frequency ratio, FR(C= +) = 0.375, is much low than that of class "–", FR(C= –) = 1. Therefore, it is reasonable to predict "–" as the class of (4, 5, ?).

From CF(C= +, N(Q,5)) = –0.25 and CF(C= –, N(Q,5)) = 0.25, it is reasonable to predict "–" as the class of (4, 5, ?).

C. Analysis

*K*NN classification is a lazy learning technique, or instance-based learning/reasoning method. Different from model-based algorithms (training models from a given dataset and then predicting a query with the models), it needs to store the training data (or cases) in memory and to compute the most relevant data to answer a given query. The answer to the query is the class represented by a majority of the k nearest neighbors. This is the majority rule. Although *k*NN classification with majority rule is simple and effective in general, there are still some limitations from an applied context, for example, cost-sensitive learning and imbalanced classification applications. Therefore, there are great many improvement efforts. We briefly discuss them from three directions as follows.

The first direction is the distance weighted *k*NN rule. Almost all improvement efforts belong to this direction. This direction is actually a selection of the k nearest neighbors for a given query. This is because different distance functions or weighting techniques (or both) can generate different k nearest neighbors only. Whatever the distance functions or weighting techniques are selected, the goal is to find a machine that highlights some attributes and decreases the impact of the rest on the query. This looks like a mapping that transforms the original space to a new space more suitable to a learning task. It is much clear when we apply the λ-cutting rule to such an algorithm. With the λ-cutting rule, the distance weighted *k*NN classification will be carried out on only those data points that the attributes are stretched out or drawn back, or a subspace consisting of attributes with the impact values equal to or greater than λ, or a combination among them.

A lately selection of the nearest neighbors is the SN (Shelly Neighbors) method that uses only those neighbors that form a shell to encapsidate the query, drawn from the k nearest neighbors [64,65]. The SN approach is actually a quadratic selection of the k nearest neighbors.

The second direction is the semi-lazy learning. This direction is actually a procedure of reducing time and space complexity. The *k*NN classification approach usually involves storing the training data in memory and completely search the training data for the k nearest neighbors. If we can properly divide the training set into n subsets and search for the k nearest neighbors from only the nearest subsets, its time and space complexity must be decreased to an acceptable computation level.

Last direction is the prediction of the query (the decision phase with the the k nearest neighbors). The usually used methods include the majority rule, weighting machine, and the Bayesian rule. The *k*NN-CF classification is a new technique that is designed against the issue of imbalanced classification.

From Section III.B, it is simple and understandable to incorporate the CF measure to *k*NN classification. It advocates to take into account the certainty factor of a classification decision when using *k*NN classification approach.

For imbalanced classification, the uncertainty is often occurred in the junction between the majority class and minority class. In this setting, the majority class certainly wins minority class in general. The Example 1 has also illustrated this uncertainty. This may lead to high cost (or risk) in many real applications, such cancer detection. The main objective of introducing the CF measure to *k*NN classification is to distinguish those classes with increased certainty factor from the classes with decreased ones.

The *k*NN-CF classification is only an idea to improve the decision phase with the the k nearest neighbors. There are some challenging issues. For example, it should be a research topic to study a new method for addressing, such as Case-1 and 2 in Figs. 2 and 3 respectively.

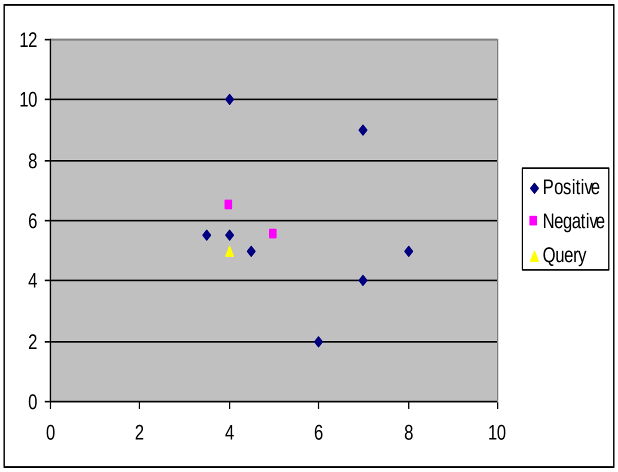

Fig. 2. Case-1 faced by the *k*NN-CF classification

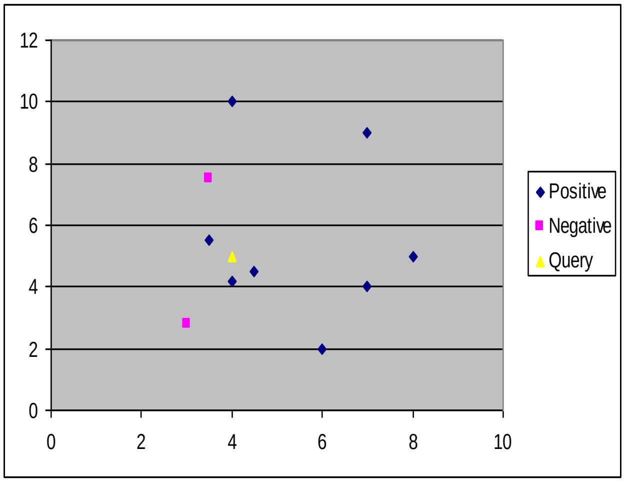

Fig. 3. Case-2 faced by the *k*NN-CF classification

From Zhang [64,65], seems the SN approach is suitable to deal with the above issues. All the above issues are studied against the joint between a minority class across and a majority. While the joint is of uncertainty, the rest are of certainty. Let ci be a majority class, cj a minority class, and Q a query. We can easily prove the following corollaries.

Corollary 1. The FR strategy is equivalent to the majority rule for kNN classification when FR(C= ci) ⩾FR(C= cj).

Corollary 2. The CF strategy is equivalent to the majority rule for *k*NN classification when CF(C= ci, N(Q,k)) ⩾CF(C= cj, N(Q,k)).

IV. Experiments

In order to show the effectiveness of the FR and CF strategies, two sets of experiments were done on real datasets with the algorithm implemented in C++ and executed using a DELL Workstation PWS650 with 2G main memory, and 2.6G CPU.

A. *Settings of experiments*

The first set of experiments was conducted for examining the efficiency against data points with pure minority class, or with pure majority class. The second set of experiments was conducted on for examining the efficiency against data points randomly drew from a dataset. Because the FR strategy is equivalent to the CF strategy, we only compare the CF strategy with Standard *k*NN approach in the following experiments. In the two sets of experiments, for simplifying the description, we always compared the proposed approaches with standard *k*NN classification. We adopt the recall and precision to evaluate the efficiecy by taking into account four distributions of minority and majority classes: 10% : 90%; 20% : 80%; 30% : 70%; 40% : 60%. For evaluating the recall and precision, all queries are randomly generated from those data points that their classes are known in a dataset. The datasets are summarized in Table II.

TABLE II

The summary of Datasets

Data set	No. of instances	Class dist. (N/P)	No. of features	No. of classes
Breast-w	683	444/239	9	2
Haberman	306	225/81	3	2
Parkinsons	195	147/48	22	2
Transfusion	748	570/178	4	2
Magic	19020	12332/6688	11	2
Ionosphere	351	225/126	33	2
Pima	768	500/268	8	2
Spambase	4601	2788/1813	57	2
SPECTF	267	212/55	44	2
wdbc	569	357/212	30	2

B. *The first group of experiments*

We examine the efficiency against data points with pure minority class, or with pure majority class. The results are showed in Tables III - VI as follows.

TABLE III

Standard *k*NN and *k*NN-CF classifications are used to predict the class of data points that are randomly generated from the known data points with the minority class for distributions: 10% : 90% and 20% : 80%

	10%:90%		20%:80%	
	kNN	kNN-CF	kNN	kNN-CF
Breast-w	80.3	93.2	92.5	96.5
Haberman	0	25.4	13.1	36.2
Parkinsons	55	73.8	79	89.5
Transfusion	5.3	16.3	25.8	45.3
Magic	38.9	53.4	53.3	68.7
Ionosphere	3.4	23.3	36.8	57.4
Pima	1.5	22.6	30	51.8
Spambase	57.9	69.4	71.4	83.6
SPECTF	6.6	38.3	23.7	75.8
wdbc	86.3	92.4	93.2	93.8
Average	33.52	50.81	51.88	69.86

TABLE IV

Standard *k*NN and *k*NN-CF classifications are used to predict the class of data points that are randomly generated from the known data points with the minority class for distributions: 30% : 70% and 40% : 60%

	30%:70%		40%:60%	
	*k*NN	*k*NN-CF	*k*NN	*k*NN-CF
Breast-w	95	98.1	97.8	98.9
Haberman	22.9	51.3	38.2	65.3
Parkinsons	85.5	94.8	87.2	95.4
Transfusion	40.2	62.9	50.1	71.3
Magic	64.1	77	70.1	81.4
Ionosphere	56.9	69	64.9	71.5
Pima	52.5	71.5	60	77.9
Spambase	79.2	88.5	87.6	92.4
SPECTF	59.6	89.6	73.9	100
wdbc	90.8	93.2	93.9	96.4
Average	64.67	79.59	72.37	85.05

TABLE V

Standard *k*NN and *k*NN-CF classifications are used to predict the class of data points that are randomly generated from the known data points with the majority class for distributions: 10% : 90% and 20% : 80%

	10%:90%		20%:80%	
	*k*NN	*k*NN-CF	*k*NN	*k*NN-CF
Breast-w	98.9	97.9	98.4	98.2
Haberman	97.9	94.1	91.5	78.2
Parkinsons	100	98	98.1	91.6
Transfusion	97.9	90.6	93.9	82.5
Magic	98.4	96	97	92.1
Ionosphere	100	98.5	97.5	97.3
Pima	98.6	92.2	93.7	82.9
Spambase	98.6	96.9	97.4	94.2
SPECTF	96.7	84	86.1	64.9
wdbc	100	99.7	99.4	98.4
Average	98.7	94.79	95.3	88.03

TABLE VI

Standard *k*NN and *k*NN-CF classifications are used to predict the class of data points that are randomly generated from the known data points with the majority class for distributions: 30% : 70% and 40% : 60%

	30%:70%		40%:60%	
	*k*NN	*k*NN-CF	*k*NN	*k*NN-CF
Breast-w	97.6	96.3	97.8	97.6
Haberman	87.3	70	75.8	51.7
Parkinsons	96	88	89.2	80.6
Transfusion	86.3	72.1	80.8	62.1
Magic	93.6	86.3	92.3	82.2
Ionosphere	98.2	97.1	97.9	97.3
Pima	89.5	75.5	81.4	64.1
Spambase	94.5	89.8	92.8	84.4
SPECTF	68.9	46.5	62.8	46.1
wdbc	99	96.2	98.2	95
Average	91.09	81.78	86.9	76.11

C. The second group of experiments

We examine the efficiency with queries randomly generated from a dataset. The results are showed in Tables VII - XIV as follows.

TABLE VII

For dataset Breastw, the efficiency of standard *k*NN and *k*NN-CF classifications when 10% : 90%

Running times	*k*NN		*k*NN-CF	
	Precision	Recall	Precision	Recall
100	0.882353	0.9375	0.882353	0.9375
200	0.75	0.882353	0.772727	1
500	0.923077	0.765957	0.851064	0.851064
1000	0.869565	0.851064	0.847619	0.946809

TABLE VII

For dataset Breastw, the efficiency of standard *k*NN and *k*NN-CF classifications when 20% : 80%

Running times	*k*NN		*k*NN-CF	
	Precision	Recall	Precision	Recall
100	1	0.894737	0.947368	0.947368
200	0.914286	0.820513	0.916667	0.846154
500	0.954545	0.903226	0.936842	0.956989
1000	0.926471	0.931034	0.908257	0.975369

TABLE IX

. For dataset Breastw, the efficiency of standard *k*NN and *k*NN-CF classifications when 30% : 70%

Running times	*k*NN		*k*NN-CF	
	Precision	Recall	Precision	Recall
100	0.866667	0.962963	0.870968	1
200	0.927536	0.955224	0.90411	0.985075
500	0.95	0.956835	0.951049	0.978417
1000	0.973244	0.960396	0.936909	0.980198

TABLE X

For dataset Breastw, the efficiency of standard *k*NN and *k*NN-CF classifications when 40% : 60%

Running times	*k*NN		*k*NN-CF	
	Precision	Recall	Precision	Recall
100	0.945946	1	0.945946	1
200	0.952381	0.987654	0.931034	1
500	0.95977	0.954286	0.935135	0.988571
1000	0.944444	0.968912	0.918465	0.992228

TABLE XI

For dataset Ionosphere, the efficiency of standard *k*NN and *k*NN-CF classifications when 10% : 90%

Running times	*k*NN		*k*NN-CF	
	Precision	Recall	Precision	Recall
100	0.888889	0.615385	0.9	0.692308
200	1	0.111111	0.9	0.5
500	1	0.1	1	0.36
1000	1	0.172414	0.782609	0.413793

TABLE XII

For dataset Ionosphere, the efficiency of standard *k*NN and *k*NN-CF classifications when 20% : 80%

Running times	*k*NN		*k*NN-CF	
	Precision	Recall	Precision	Recall
100	1	0.8	1	0.866667
200	0.884615	0.469388	0.9	0.734694
500	0.933333	0.482759	0.944444	0.586207
1000	0.666667	0.134715	0.801802	0.46114

TABLE XIII

For dataset Ionosphere, the efficiency of standard *k*NN and *k*NN-CF classifications when 30% : 70%

Running times	*k*NN		*k*NN-CF	
	Precision	Recall	Precision	Recall
100	0.931034	0.72973	0.916667	0.891892
200	0.939394	0.563636	0.944444	0.618182
500	0.949367	0.517241	0.948454	0.634483
1000	0.920455	0.514286	0.932039	0.609524

TABLE XIV

For dataset Ionosphere, the efficiency of standard *k*NN and *k*NN-CF classifications when 40% : 60%

Running times	*k*NN		*k*NN-CF	
	Precision	Recall	Precision	Recall
100	1	0.657895	1	0.684211
200	0.959184	0.580247	0.967742	0.740741
500	0.971429	0.676617	0.943396	0.746269
1000	0.95082	0.659091	0.939481	0.740909

From Tables III-XIV, the *k*NN-CF is much better than standard *k*NN classification at predicting the minority class. This indicates that the CF strategy is promising to reduce the misclassification cost for real applications, such as disease diagnosis and risk-sensitive learning.

V. CONCLUSIONS AND OPEN PROBLEMS

In this paper we have incorporated the certainty factor to *k*NN classification that clearly distinguishes whether the belief of the class of a query is increased, given its k nearest neighbors. We have experimentally illustrated the efficiency of the proposed approach, *k*NN-CF classification. For future study, we list some open problems in *k*NN-CF classification as follows.

1. Improve the discernment of *k*NN-CF classification with a means, such as the SN approach in [64,65].
2. Extending the *k*NN-CF classification to cost/risk-sensitive learning.
3. *k*NN-CF classification with missing values.
4. *k*NN-CF classification with cold-deck instances [38].
5. The evaluation of *k*NN-CF classification.

ACKNOWLEDGMENT

I am grateful for both the suggestions and the experiments carried out by my ex-student, Mr Manlong Zhu. Many thanks for the comments on the early version of this paper from Mr Xiaofeng Zhu (my ex-student), and Dr. William K. Cheung.

This work was supported in part by the Australian Research Council (ARC) under grant DP0985456, the Nature Science Foundation (NSF) of China under grant 90718020, and the Guangxi NSF (Key) grants.

REFERENCES

[1] D.W. Aha, D.F. Kibler and M.K. Albert (1991). Instance-based learning algorithms. *Machine Learning*, Vol. 6: 37–66.

[2] V. Athitsos, J. Alon, S. Sclaroff and G. Kollios (2008). BoostMap: An Embedding Method for Efficient Nearest Neighbor Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell*. 30(1): 89-104.

[3] S. Belongie, J. Malik, and J. Puzicha (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell*., 24(4): 509-522.

[4] E. Blanzieri and F. Melgan (2008). Nearest Neighbor Classification of Remote Sensing Images With the Maximal Margin Principle. *IEEE Trans.Geoscience and Remote Sensing*, 46(6): 1804-1811.

[5] E. Blanzieri and F. Ricci (1999). Probability Based Metrics for Nearest Neighbor Classification and Case-Based Reasoning. *Lecture Notes in Computer Science*, Vol. 1650: 14-29.

[6] D.R. Carvalho and A.A. Freitas (2002). A genetic-algorithm for discovering small-disjunct rules in data mining. *Appl. Soft Comput*., Vol. 2(2): 75–88.

[7] D. Carvalho and A. Freitas (2004). A hybrid decision tree/genetic algorithm method for data mining. *Inf. Sci*., Vol. 163(1-3): 13–35.

[8] J. Chai, H. Liu, B. Chen and Z. Bao (2010). Large margin nearest local mean classifier. *Signal Processing*, 90: 236-248.

[9] N.V. Chawla, K.W. Bowyer, L.O. Hall and W.P. Kegelmeyer (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16: 321–357.

[10] Chen, J., and Shao, J., (2001). Jackknife variance estimation for nearest-neighbor imputation. *J. Amer. Statist. Assoc*. 2001, Vol. 96: 260-269.

[11] Cheung, K., Fu, A. (1998). Enhanced Nearest Neighbour Search on the R-tree. *SIGMOD Record*, Vol 27: 16-21.

[12] T. Cover and P. Hart (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, Vol. 13: 21–27.

[13] Daugman, J. (2003). Demodulation by Complex-Valued Wavelets for Stochastic Pattern Recognition. Int’l J. Wavelets, *Multiresolution and Information Processing*, pp 1-17.

[14] Dietterich, T., and Sutton, R. (1995). An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning*, 19: 5-28.

[15] C. Domeniconi and D. Gunopulos (2001). Adaptive nearest neighbor classification using support vector machines. In *NIPS*, pp 665-672.

[16] C. Domeniconi, J. Peng and D. Gunopulos (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(9): 1281-1285.

[17] P. Domingos (1998). How to get a free lunch: A simple cost model for machine learning applications. *Proc. AAAI98/ICML98, Workshop on the Methodology of Applying Machine Learning. AAAI Press*, pp. 1–7.

[18] P. Domingos (1999). Metacost: A general method for making classifiers cost-sensitive. *In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pp. 155–164.

[19] C. Elkan (2001). The foundations of cost-sensitive learning. *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pp. 973–978.

[20] T. Fawcett and F. J. Provost (1997). Adaptive fraud detection. *Data Min. Knowl. Discov*., 1(3): 291–316.

[21] Fix E, Hodges JL, Jr (1951). Discriminatory analysis, nonparametric discrimination. USAF School of Aviation Medicine, Randolph Field, Tex., Project 21-49-004, Rept. 4, Contract AF41(128)-31, February 1951.

[22] B. J. Frey and D. Dueck (2007). Clustering by Passing Messages Between Data Points. Science, 315: 972-976.

[23] Guo, G., Wang, H., Bell, D., Bi, Y., and Greer, K. (2003). KNN Model-Based Approach in Classification. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pp 986-996.

[24] T. Hastie and R. Tibshirani (1996). Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Anal. Mach. Intell*., 18(6): 607-615.

[25] R.C. Holte, L. Acker and B.W. Porter (1989). Concept learning and the problem of small disjuncts. *IJCAI-89*, pp. 813–818.

[26] T.M. Huard and S. Robin (2009). Tailored Aggregation for Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11): 2098-2105.

[27] P. Jing, D.R. Heisterkamp and H.K. Dai (2001). LDA/SVM driven nearest neighbor classification. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog*., pp. 58-63.

[28] M. Kubat, R. Holte, and S. Matwin (1998). Machine learning for the detection of oil spills in satellite radar images. *MachineLearning*, 2-3, pp. 195–215.

[29] M. Kubat and S. Matwin (1997). Addressing the curse of imbalanced training sets: One-sided selection. *In Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 179–186.

[30] W. Lam and Y. Han (2003). Automatic Textual Document Categorization Based on Generalized Instance Sets and a Metamodel. *IEEE Trans. Pattern Anal. Mach. Intell*., 25(5): 628-633.

[31] B. Li, Y.W. Chen and Y. Chen (2008). The Nearest Neighbor Algorithm of Local Probability Centers. *IEEE Trans. Syst., Man, Cybern. (B)*, 38(1): 141-153.

[32] C. Ling and C. Li (1998). Data mining for direct marketing: Problems and solutions. *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 73–79.

[33] T. Menzies, J. Greenwald and A. Frank (2007). Data mining static code attributes to learn defect predictors. *IEEE Transactionson Software Engineering*, 33: 2–13.

[34] Mitani, Y. and Hamamoto, Y. (2006). A local mean-based nonparametric classifier. *Pattern Recognition Letters*, 27(10): 1151-1159.

[35] K. Ni and T. Nguyen (2009). An Adaptable k-Nearest Neighbors Algorithm for MMSE Image Interpolation. *IEEE Transactions on Image Processing*, 18(9): 1976-1987.

[36] V. Pascal and B. Yoshua (2003). Manifold Parzen windows. *Proc. NIPS*, pp. 825-832.

[37] J. Peng, D. Heisterkamp and H.K. Dai (2004). Adaptive Quasiconformal Kernel Nearest Neighbor Classification. *IEEE Trans. Pattern Analysis and Machine lntelligence*, 26(5): 656-661.

[38] YS Qin, SC Zhang (2008). Empirical Likelihood Confidence Intervals for Differences between Two Datasets with Missing Data. *Pattern Recognition Letters*, Vol 29(6): 803-812.

[39] J.R. Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.

[40] J.R. Quinlan, P.J. Compton, K.A. Horn and L. Lazarus (1986). Inductive knowledge acquisition: a case study. *Proceedings of the second Australian Conference on the Applications of Expert Systems*, pp. 183–204.

[41] S. Salzberg (1991). A Nearest Hyperrectangle Learning Method. *Machine Learning*, Vol. 6: 251-276.

[42] H. Samet (2008). K-Nearest Neighbor Finding Using MaxNearest-Dist. *IEEE Trans. Pattern Anal. Mach. Intell*. 30(2): 243-252.

[43] Sanchez, J.S., Pla, F. and Ferri, F. J. (1997). On the use of neighbourhood based non-parametric classifiers. *Pattern Recognition Letters*, 18: 1179-1186.

[44] P. Simard, Y. LeCun and J.S. Denker (1993). Efficient pattern recognition using a new transformation distance. *Proceedings of NIPS*, pp 50-58.

[45] Singh, S., Haddon, J., Markou, M. (1999). Nearest Neighbour Strategies for Image Understanding. *Proc. Workshop on Advanced Concepts for Intelligent Vision Systems (ACIVS'99)*, Baden-Baden, pp 2-7.

[46] Shortliffe, E. and Buchanan, B. (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23: 351–379.

[47] C. Stanfill and D. Waltz (1986). Toward Memory-Based Reasoning. Comm. ACM, Vol. 29: 1213-1229.

[48] Y. Sun, M.S. Kamel, A. Wong and Y. Wang (2007). Costsensitive boosting for classification of imbalanced data. *Pattern Recogn.*, Vol. 40(12): 3358–3378.

[49] Tao, Y., Papadias, D., Lian, X. (2004). Reverse knn search in arbitrary dimensionality. *VLDB-2004*, pp 744-755.

[50] J.B. Tenenbaum and V. de Silva and J.C. Langford (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290: 2319-2323.

[51] K. Ting (1994). The problem of small disjuncts: its remedy in decision trees. *In Proceedings of the Tenth Canadian Conference on Artificial Intelligence*, pp. 91–97.

[52] M. Varma and A. Zisserman (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2): 61-.81.

[53] P. Vincent and Y. Bengio (2001). K-local hyperplane and convex distance nearest neighbor algorithms. *In NIPS*, pp 985-992.

[54] H. Wang (2006). Nearest neighbors by neighborhood counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28: 942-953.

[55] G.M. Weiss (2004). Mining with rarity: a unifying framework. *SIGKDD Explorations*, 6(1): 7–19.

[56] G.M. Weiss and H. Hirsh (2000). A quantitative study of small disjuncts. *AAAI/IAAI*, pp. 665–670.

[57] G. Wen, L. Jiang and J. Wen (2008). Using Locally Estimated Geodesic Distance to Optimize Neighborhood Graph for Isometric Data Embedding. *Pattern Recognition*, 41: 22-26.

[58] G. Wen, et al. (2009). Local relative transformation with application to isometric embedding. *Pattern Recognition Letters*, 30(3): 203-211.

[59] D.R. Wilson and T.R. Martinez (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, 383, pp. 257–286, Kluwer Academic Publishers.

[60] Wu, XD., et al. (2008). Top 10 Algorithms in Data Mining, Knowledge and Information Systems, 14(1): 1-37.

[61] Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, Vol 1: 67-88.

[62] Y. Zeng, Y. Yang and L. Zhao (2009). Nonparametric classification based on local mean and class statistics. *Expert Systems with Applications*, 36: 8443-8448.

[63] H. Zhang, A. Berg, M. Maire and J. Malik (2006). SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. *Proceeding of CVPR-2006*, pp 2126-2136.

[64] Zhang, SC. (2008). Parimputation: From imputation and null-imputation to partially imputation. *IEEE Intelligent Informatics Bulletin*, Vol 9(1), 2008: 32-38.

[65] Zhang, SC. (2010). Shell-Neighbor Method And Its Application in Missing Data Imputation. *Applied Intelligence, DOI*: 10.1007/s10489-009-0207-6.

[66] Zhang, S.C., Qin, Z.X., Sheng, S.L. and Ling, C.L. (2005). “Missing is useful”: Missing values in cost-sensitive decision trees. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17 No. 12: 1689-1693.

[67] Zhang, S.C., Zhang, C.Q. and Yang, Q. (2004). Information Enhancement for Data Mining. *IEEE Intelligent Systems*, March/April 2004: 12-13.

[68] H. Zhu and O. Basir (2005). An Adaptive Fuzzy Evidential Nearest Neighbor Formulation for Classifying Remote Sensing Images. *IEEE Trans. Geoscience and Remote Sensing*, 43(8): 1874-1889.

[69] J. Zhu (2007). Active learning for word sense disambiguation with methods for addressing the class imbalance problem. *In Proceedings of ACL*, pp. 783–790.

[70] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. *Proceedings of Computer Vision and Pattern Recognition(2005)* , volume 1, pages 26–33, 2005.

[71] X.Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules.*ACM Transactions on Information Systems*, 22(3):381 – 405, July 2004.

[72] S. Zhang and X. Wu. Fundamental Role of Association Rules in Data Mining and Knowledge Discovery. *WIREs Data Mining & Knowledge Discovery*, 2011

[73] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, Building useful models from imbalanced data with sampling and boosting. *In Proceedings of 21st Int. FLAIRS Conference,* May 2008, pp. 306–311.