A Hybrid Coupled k-Nearest Neighbor Algorithm on Imbalance Data

Chunming Liu[1], Longbing Cao[1] and Philip S Yu[2]
[1]Advanced Analytics Institute, University of Sydney Technology, Australia
Chunming.Liu@student.uts.edu.au, LongBing.Cao@uts.edu.au
[2]Computer Science, University of Illinois at Chicago, USA
psyu@cs.uic.edu

***Abstract*— The state-of-the-art classification algorithms rarely consider the relationship between the attributes in the data sets and assume the attributes are independently to each other (IID). However, in real-world data, these attributes are more or less interacted via explicit or implicit relationships. Although the classifiers for class-balanced data are relatively well developed, the classification of class-imbalanced data is not straightforward, especially for mixed type data which has both categorical and numerical features. Limited research has been conducted on the class-imbalanced data. Some algorithms mainly synthesize or remove instances to force the sizes of each class comparable, which may change the inherent data structure or introduces noise to the source data. While for the distance or similarity based algorithms, they ignored the relationship between features when computing the similarity. This paper proposes a hybrid coupled k-nearest neighbor classification algorithm (HC-kNN) for mixed type data, by doing discretization on numerical features to adapt the inter coupling similarity as we do on categorical features, then combing this coupled similarity to the original similarity or distance, to overcome the shortcoming of the previous algorithms. The experiment results demonstrate that our proposed algorithm can get a higher average performance than that of the relevant algorithms (e.g. the variants of kNN, Decision Tree, SMOTE and NaiveBayes).**

I. Introduction

Classification analysis plays an important practical role in several domains, such as machine learning and data mining. Classification techniques have been widely used in retail, finance, banking, security, astronomy, and behavioral ecology, etc. [1].

In many research and application areas, data sets could be a mixture of categorical and numerical attributes (mixed data sets). If the objects are described by numerical attributes, their similarity measures reflect the direct relationship between data values. For example, the values pair (100kg, 120kg) are more similar than (100kg, 20kg), in other words, more close to each other. A variety of similarity metrics have been developed for numerical data, such as Euclidean and Minkowski distances. While with categorical data, although several similarity measures, such as the Jaccard coeffcient [2], overlap, and Goodall similarity [3] can be used, they are usually not as straightforward and general as similarities for continuous data.

The classification analysis on the class-imbalanced datasets has received much less attention, especially for the mixed type data described by numerical and categorical features. It has been observed that the traditional algorithms do not perform as good on imbalanced datasets as on balanced datasets. In the literature of solving class imbalance problems, various solutions have been proposed. In general, all these methods can be broadly divided into two different approaches: data re-sampling and modifying existing methods.

Although sampling-based methods show to outperform the original algorithms in most situation, they do not introduce much improvement for kNN, especially on imbalanced categorical data. This may be partly explained by the maximum-specificity induction bias of kNN in which the classification decision is made by examining the local neighbourhood of query instances, and therefore the global re-sampling strategies may not have pronounced effect in the local neighbourhood under examination. In addition, re-sampling strategies inevitably change the inherent relationships of the original data, or even worse, lose information or add noise. In dealing with class-imbalanced classification tasks, some distance or similarity-based classification algorithms are proposed, such as kENN[4] and CCW-kNN[5]. However, they do not consider the relationship between the features when they compute the similarity/distance between instances. We illustrate the problems with the existing work and highlight the challenge of classifying class-imbalanced mixed type data below.

Taking some of the UCI Nursery data (Table I) as an example, eleven instances are divided into two classes with four categorical features: parents, has-nurs, form and social. The value in the brackets indicates the frequency of the corresponding feature value. It is a class-imbalanced categorical data set. Here, we use the first instance $\{u_0\}$ as the testing data set, and the rest $\{u_i\}_{i=1}^{10}$ as the training data set. If we use the traditional kNN algorithm to classify u_0, it will be labeled as B due to a relatively large number of the instances in class B. As shown in Table I, the Overlap Similarity, which is defined as

$$\text{Sim\_Overlap} = \frac{|A \bigcap B|}{\min\{|A|, |B|\}}, \tag{1}$$

the maximum similarity is $Sim\_Overlap(u_0, u_4)$, which is 0.75. If we adopt the Cosine Similarity, which is defined as

$$\text{Sim\_Cosine} = \frac{A \cdot B}{||A||\,||B||}, \tag{2}$$

TABLE I
AN EXAMPLE FROM THE UCI DATASET: NURSERY DATA

ID	parents	has-nurs	form	social	Class	Overlap Similarity	Cosine Similarity
u_0	usual	improper	foster	nonprob	A		
u_1	usual (4)	proper (4)	incomplete (4)	slightly-prob (5)	A	0.25	0.8484
u_2	pretentious (4)	less-proper (3)	completed (2)	nonprob (2)	A	0.25	0.9278
u_3	usual (4)	less-proper (3)	incomplete (4)	slightly-prob (5)	B	0.25	0.8660
u_4	usual (4)	improper (1)	incomplete (4)	nonprob (2)	B	0.75	0.8762
u_5	usual (4)	critical (1)	completed (2)	problematic (3)	B	0.25	0.9731
u_6	pretentious (4)	proper (4)	complete (3)	problematic (3)	B	0	0.8744
u_7	pretentious (4)	proper (4)	incomplete (4)	slightly-prob (5)	B	0	0.8484
u_8	pretentious (4)	less-proper (3)	foster (1)	slightly-prob (5)	B	0.25	0.8956
u_9	great-pret (2)	proper (4)	complete (3)	slightly-prob (5)	B	0	0.7253
u_{10}	great-pret (2)	very-crit (1)	complete (3)	problematic (3)	B	0	0.8002

then the instances u_5, u_2, u_8 and u_4 will be the top 4 instances which are close to u_0. Under this scenario, u_0 will be assigned to class B rather than class A no matter what k we choose in kNN, because there are always more nearest neighbors labeled as class B than as class A.

The main problem of categorical data is that there are no inherent order in the different values that a categorical attribute takes. We can not tell whether the word "Cloudy" is in the middle of the words "Sunny" and "Rainy" or not. Thus, it is not possible to directly compare two different categorical values. People using the simplest way, the overlap measure, to find similarity between two categorical attributes. It assigns a 1 if the values are identical and a 0 if the values are not identical. Then for two multivariate categorical data points, the similarity between them will be expressed by the number of attributes in which they match, as shown in Equation 1. The overlap measure does not distinguish between the different values taken by an attribute, all matches as well as mismatches, are treated as equal (and assign a value 1). This will cause problems in some situations. For example, considering a categorical data set D, which has only two features: weather and time. Weather takes three possible values: Sunny, Cloudy, Rainy, and time takes three values: morning, afternoon and evening. Table II shows the frequency of co-occurrence of the two features.

Based on data set D, the overlap similarity between the two instances (Cloudy,morning) and (Cloudy,afternoon) is $\frac{1}{2}$, and the overlap similarity between (Rainy,morning) and (Rainy,afternoon) is also $\frac{1}{2}$. But the frequency distribution in Table II shows that the first pair are frequent combinations, while the second pair are very rare combinations in the data set. Hence, the overlap measure is too simplistic to give equal importance to all matches and mismatches. This example shows that there is some other information in categorical data sets that can be used to define what makes two values more or less similar.

In computing the cosine similarity of categorical data, the vector comes from the frequency of a single value of a feature, so it ignores the information hiding in the co-occurrence of two features.

These examples show that traditional classification algorithms are unable to capture the genuine relationships between imbalanced classes and between features. Learning from the class-imbalanced data has also been identified as one of the top 10 challenging problems in data mining research [6].

In this paper, we propose a novel hybrid coupled nearest neighbor classification algorithm for class-imbalanced mixed type data by addressing both the relationships between classes and between features. The key contributions are as follows:

- We assign the corresponding size memberships to distinct classes according to their sizes to handle the class-imbalanced issue in a fuzzy way.
- We extend the coupled relationship to numerical features by using discretization techniques.
- We explore the coupled interactions within each feature and between different features to produce a relatively more accurate similarity measurement between mixed-type instances.
- We compare the performance of our proposed algorithm with existing methods on the ROC curve, and the results confirm the improvement.

The paper is organized as follows. Section II briefly reviews the related work. Preliminary definitions are specified in Section III. Section IV explains our classification algorithm on the class-imbalanced data sets. The experimental results are discussed in Section V. The conclusion and future work are summarized in Section VI.

II. RELATED WORK

In dealing with class imbalance classification problems, many solutions have been proposed. In general, all these methods can be broadly divided into three different approaches: data sampling, algorithmic modification and cost-sensitive learning[7]. The data sampling methods focus on balancing the data, and the common strategies are to reduce the majority class examples (undersampling) or to add new minority class examples to the data (oversampling)[8], [9]. One of the most famous over-sampling methods is SMOTE[8]. It over-samples the minority class by taking each minority class sample and introducing synthetic examples along the line segments joining all of the k minority class nearest neighbors, so it is also based on the nearest neighbor analogy. It beats the random over-sampling by adding new instances to a minority class, without suffering from the over-fitting.

TABLE II
FREQUENCY OF FEATURE CO-OCCURRENCE

	morning	*afternoon*	*evening*	**Total**
Sunny	44	47	9	100
Cloudy	48	45	7	100
Rainy	8	8	84	100
Total	100	100	100	

Unlike our focus here, the methods of these re-sampling are designed more suitable for numerical data sets. SMOTE would introduce noise points if it is used for categorical data. Unlike re-sampling methods which change the original data structure, the approaches modifying existing algorithms alter the existing classification algorithms to make them more effective in dealing with imbalanced data, while keeping the data structure unchanged. For example, CCPDT[10], which is designed for imbalanced situation, is a modification of the decision tree algorithm. The cost-sensitive learning incorporate approaches at the data level, algorithmic level or at both levels, considering higher costs for the misclassification of examples of the positive class with respect to the negative class, and trying to minimize higher cost errors[11].

Although kNN has been identified as one of the top ten most influential data mining algorithms, the standard kNN algorithm is not suitable for the presence of imbalanced class distribution. To improve the performance of kNN for imbalanced classification, kENN[4] and CCW-kNN[5] have been proposed. kENN proposed a training stage where exemplar positive training instances are identified and generalized into Gaussian balls as concepts for the minority class. When classifying a query instance using its k nearest neighbors, the positive concepts formulated at the training stage ensure that classification is more sensitive to the minority class. This approach is based on extending the decision boundary for the minority class. CCW-kNN uses the probability of attribute values given class labels to weight prototypes in kNN. They used conditional probabilities of classes but not the probabilities of class labels in the neighborhood of the query instance. These methods perform more accurately than the existing algorithms. However both kENN and CCW-kNN require a training stage either to find exemplar training samples to enlarge the decision boundaries for the positive class, or to learn the class weight for each training sample by mixture modelling and Bayesian network learning. The computational cost can be substantial for both approaches, which are more suitable for numerical data.

Yang Song et al. [12] propose two new kNN algorithms based on informativeness, which is introduced as a query-based distance metric. A point is treated informative if it is close to the query point and far away from the points with different class labels. Locally Informative kNN(LI-kNN) applies this to select the most informative points and predict the label of a query point based on the most numerous class with the neighbors; Globally Informative kNN(GI-kNN) finds the globally informative points by learning a weight vector from the training points.

The above introduces new learning algorithms to deal with the imbalanced class distribution problem, but they focus on handling numerical data. The overlap similarity or cosine similarity[13] for categorical data is too vague to clearly describe how close two categorical instances are. Those similarity measures assume that the categorical features are independent to each other. However, more researchers argue that the similarity between categorical feature values is also dependent on the couplings of other features [3]. Wang et al. [14] presents a coupled nominal similarity to examine both the intra-coupling and inter-coupling of categorical features. Their approaches focus on the clustering learning on the class-balanced data; whereas our proposed method considers the classification learning on the class-imbalanced categorical data, which has not been systematically addressed so far.

III. PROBLEM STATEMENT

Classification learning on the class-imbalanced categorical data can be formally described as follows: $U = \{u_1, \cdots, u_m\}$ is a set of m instances; $A = \{a_1, \cdots, a_n\}$ is a set of n categorical and numerical features; $C = \{c_1, \cdots, c_L\}$ is a set of L classes, in which each class has dramatically different numbers of instances. The goal is to classify an unlabeled testing instance u_t based on the instances in the training set $\{u_i\}$ with known classes. For example, Table I exhibits a class-imbalanced data set. The training set consists of ten objects $\{u_1, u_2, \cdots, u_{10}\}$, four features $\{parents, hasnurs, form, social\}$, and two classes $\{A, B\}$. There are only two instances in class A, while eight instances in class B. Our task is to find a suitable classification model to categorize u_0 into class A.

In the following sections, the size of a class refers to the number of instances in this class. When we say a class c_l is smaller (or larger) than c_k, it means that the size of class c_l is smaller (or larger) than that of c_k. A minority class has a relatively small size, while a majority class has a relatively large size. In addition, $|H|$ is the number of instances in set H.

IV. HYBRID COUPLED CLASSIFICATION

In this section, a hybrid coupled kNN algorithm (i.e. *HC-kNN* for short) is proposed to handle the classification problem on the class-imbalanced mixed type data sets. Algorithm 1 illustrates the main idea of our algorithm.

HC-kNN consists of five parts: *membership assignment*, *data discretization*, *feature weighting*, *similarity calculation*, and *integration*. At the phase of membership assignment, we introduce a fuzzy membership to handle the class-imbalanced issue: *Sized Membership of Class*. This membership provides the quantification on how small a class is. At the step of data discretization, we use CAIM discretization algorithm [15] which can capture the class-attribute interdependency information on numerical features. In the third part of feature weighting calculation, we use feature-class coupled relationship to assign every feature a proper weight. At the step of similarity calculation, we present the *Adapted Coupled Nominal Similarity* to describe the closeness of two different instances. Finally, at the final stage of integration, we propose the *Integrated Similarity* to measure the similarity between

Algorithm 1 : Hybrid Coupled kNN Algorithm

Input: An instance u_t without label and a source labeled dataset $D\{u_1, u_2, ..., u_n\}$

Output: The class label of u_t

1: For each class, initiate the sized membership of class using the fuzzy set theory
2: Do discretization on numerical features
3: Calculate the feature weight of every feature
4: Create the similarity matrix which contains both intra and inter similarity for dataset D
5: Calculate the distance of u_t to every instance in dataset D using the adapted similarity
6: Select top k points which are close to the instance u_t
7: Return the class label of those k neighbors which has the maximum number of instances

the test instance and the training instance by merging the adapted coupled nominal similarity and fuzzy membership of a class. The classification result of a test instance is determined according to the integrated pairwise similarity. Below, we specify all the building blocks one by one.

A. Membership Assignment

In this part, we propose a membership: *Sized Membership of Class* to characterize the structure of imbalanced classes and to capture the prior knowledge integrated from the instances.

In the class-imbalanced data set, there are usually several small classes that contain much less instances (i.e. minority), while a lot more instances are in some large classes (i.e. majority). However, what exactly does a small class mean? How do we quantify a small class? As it would be too reductive to regard the smallest class as the minority, we use a fuzzy way [16] to measure how small a class is according to its size. Accordingly, we have:

Definition 1: The **Sized Membership of Class** $\theta(\cdot)$ denotes the rate of a class c_l that belongs to the minority. Formally, $\theta(\cdot)$ is defined as:

$$\theta(c_l) = 1 - \frac{|c_l|}{m}, \tag{3}$$

where $|c_l|$ is the number of instances in classes c_l and m is the total number of instances in the data set. Accordingly, we have $\theta(c_l) \in (0, 1)$.

The sized membership of class describes how small a class is. In special cases, $\theta(c_l)$ reaches the maximum if c_l has the smallest number of instances; $\theta(c_l)$ is down to the minimum if c_l is the largest class. For other medium classes, the corresponding sized membership of class falls within $(\theta(c_l)^{min}, \theta(c_l)^{max})$. When a data set is balanced with two classes, where we have $\theta(c_l) = 0.5$. In Table I, for instance, we have $\theta(c_A) = 1 - 2/10 = 4/5$, and $\theta(c_B) = 1 - 8/10 = 1/5$.

Later in measuring the similarity of instances, we will incorporate the sized membership of class $\theta(\cdot)$ into the integrated similarity measure to balance the impact of class size in measuring instance similarity.

B. Data Discretization

In order to apply our strategy which compute the similarity between numerical features and categorical features, we do discretization on numerical attributes to transfer such continues values into separate groups. As we are conducting the supervised classification tasks, we choose CAIM (class-attribute interdependence maximization) discretization algorithm [15] which can capture the class-attribute interdependency information as our discretization method.

The algorithm uses class-attribute interdependency information as the criterion for the optimal discretization. For a given quanta matrix, the CAIM criterion measures the dependency between the class variable C and the discretization variable D for attribute F. It is defined as:

$$\text{CAIM}(C, D|F) = \frac{\sum_{r=1}^{n}(max_r^2/M_{+r})}{n}, \tag{4}$$

where n is the number of intervals, r iterates through all intervals, and max_r is the maximum value within the rth column of the quanta matrix, M_{+r} is the total number of continuous values of attribute F that are within the interval $(d_{r-1}, d_r]$.

The algorithm starts with a single interval that covers all possible values of a continuous attribute, and divides it iteratively. From all possible division points that are tried it chooses the division boundary that gives the highest value of the CAIM criterion.

The result we got from this discretization on numerical attributes is only used in the following Feature Weighting stage and Inter-similarity calculation stage. The reason is that we cannot compute the similarity between a numerical value and a categorical value directly for the numerical value is continues. So we use the discretization intervals as the categories of the continues values, then we can evaluate the similarity between numerical data and categorical data.

C. Feature Weighting

Definition 2: The **feature weight** describes the importance degree of each categorical feature / discretized numerical feature f_j according to its value distribution consistency with the distribution of classes. Formally, we have:

$$\alpha_j = \begin{cases} \sum_{i=1}^{m} \frac{Fre(x_{ij}, R^{C(u_i)})}{m \cdot |R^{C(u_i)}|} & \text{if} \quad |Unique(f_j)| > 1 \\ 0 & \text{if} \quad |Unique(f_j)| = 1 \end{cases} \tag{5}$$

where m is the total number of instances in the data set, x_{ij} is the j feature value for instance u_i, $R^{C(u_i)}$ consists of all the instances which share the same class as instance u_i, and the according instance number is $|R^{C(u_i)}|$, while $Fre(x_{ij}, R^{C(u_i)})$ defines as a frequency count function that count the occurrences of x_{ij} in feature j of set $R^{C(u_i)}$, and $|Unique(f_j)|$ returns the category number or discretization interval number in feature j.

The weight α_j indicates the distribution matching degree of the values of a feature to the class labels. For example, if a training data set has 6 instances with class labels of $\{C_1, C_1, C_2, C_2, C_2, C_1\}$ respectively, and feature f_1 has values of $\{A, A, B, B, B, A\}$ while feature f_2 has values of $\{M, M, M, N, N, N\}$, then the value distribution of feature f_1 is more consistent with the distribution of classes than f_2 does. The more consistent in distribution for the feature values to the class labels, the more important the feature is. If all the values in a feature are the same, that is, $|Unique(f_j)| = 1$, then this feature cannot be used in the classification task so we set the weight to be zero. We also regard this feature weight as *the coupling relationship between features and labels*. For example, in Table I, we will have the normalized feature weights: $\alpha_1 = 0.2586, \alpha_2 = 0.2069, \alpha_3 = 0.2414$, and $\alpha_4 = 0.2931$.

D. Similarity Calculation

In this part, the similarity between instances is defined for the class-imbalanced data. The usual way to deal with the similarity between two categorical instances is the cosine similarity on frequency and overlap similarity on feature category. However, they are too rough to measure the similarity and they do not consider the coupling relationships among features. Wang et al. [14] introduce a coupled nominal similarity (COS) for categorical data, which addresses both the intra-coupling similarity within a feature and the inter-coupling similarity among different features. The proposed similarity measure has been shown to outperform the SMS and the ADD[17] in the clustering learning. Here, we adapt the COS in our classification algorithm and extend it to mixed type data which contains both categorical features and numerical features. We use the Euclidean distance in our intra-similarity calculation on numerical features, and if the inter-similarity calculation relats to numerical features, we apply a same strategy on its discretization result as we do on categorical features.

Definition 3: Given a training data set D, a pair of values $v_j^x, v_j^y (v_j^x \neq v_j^y)$ of feature a_j. v_j^x and v_j^y are defined to be intra-related in feature a_j. The **Intra Coupled Similarity** (IaCS) between categorical feature values v_j^x and v_j^y of feature a_j is formalized as:

$$\delta^{Ia}(v_j^x, v_j^y) = \frac{RF(v_j^x) \cdot RF(v_j^y)}{RF(v_j^x) + RF(v_j^y) + RF(v_j^x) \cdot RF(v_j^y)}, \tag{6}$$

where $RF(v_j^x)$ and $RF(v_j^y)$ are the relative occurrence frequency of values v_j^x and v_j^y in feature a_j, respectively.
The Intra Coupled Similarity just reflects the interaction of two values in the same feature. The higher these frequencies are, the closer such two values are. Thus, Equation (6) is designed to capture the value similarity in terms of occurrence times by taking into account the frequencies of categories. Besides, since $1 \leq RF(v_j^x), RF(v_j^y) \leq m$, then $\delta^{Ia} \in [1/3, m/(m+2)]$. For example, in Table I, values "usual" and "great-pret" of feature $parents$ are observed four and two times, so $\delta^{Ia}((usual), (great - pret)) = (4 * 2)/(4 + 2 + 4 * 2) = 4/7$.

For numerical features, we use $1/Euclidean$ as the feature values' Intra-similarity δ^{Ia}.

In contrast, the Inter Coupled Similarity below is defined to capture the interaction of two values (or the group in the discretization result) from two different features.

Definition 4: Given a training data set D and two different features a_i and a_j $(i \neq j)$, two feature values $v_i^x, v_j^y (i \neq j)$ from features a_i and a_j, respectively. v_i^x and v_j^y are defined to be inter-related if there exists at least one pair value (v_p^{xy}) that co-occurs in features a_i and a_j of instance U_p. The **Inter Coupled Similarity** (IeCS) between feature values v_i^x and v_j^y of feature a_i and a_j is formalized as:

$$\delta^{Ie}_{i|j}(v_i^x, v_j^y) = \frac{F(v_p^{xy})}{\max(RF(v_i^x), RF(v_j^y))}, \tag{7}$$

where $F(v_p^{xy})$ is the co-occurrence frequency count function with value pair v_p^{xy}, and $RF(v_i^x)$ and $RF(v_j^y)$ is the relative occurrence frequency in their features respectively.

Accordingly, we have $\delta^{Ie}_{i|j} \in [0, 1]$. The Inter-Coupled Similarity reflects the interaction or relationship of two categorical values from two different features. In Table I, for example, as $\delta^{Ie}_{1|4}((usual), (problematic)) = 1/\max(4, 3) = 0.25 < \delta^{Ie}_{1|4}((great\text{-}pret), (problematic)) = 1/\max(2, 3) = 0.667$, so between feature 1 (parents) and feature 4 (social), the value pair [(great-pret),(problematic)] is more close to each other than the value pair [(usual),(problematic)].

Though the superiority of COS has been verified for clustering, there is no evidence showing that it still works well in classification, due to its lack of class information. Hence, we need to work out an adapted strategy to incorporate the classes into COS via the following feature weighting. First, the correspondence problem in relation to mapping between the feature values and the classes needs to be solved. The optimal correspondence can be obtained by using the Hungarian method with $O((n_j)^3)$ complexity for n_j feature values. Below, the correspondence mapping is built for each feature a_j $(1 \leq j \leq n)$ and a set of classes C.

By taking into account the feature importance, the *Adapted Coupled Object Similarity* between instances u_{i_1} and u_{i_2} is formalized as:

$$\begin{aligned} &AS(u_{i_1}, u_{i_2}) \\ &= \sum_{j=1}^{n} [\beta \cdot \alpha_j \delta_j^{Ia} + (1-\beta) \cdot \sum_{k=1, k\neq j}^{n} \delta^{Ie}_{j|k}] \\ &= \sum_{j=1}^{n} [\beta \cdot \alpha_j \delta_j^{Ia}(v_j^{i_1}, v_j^{i_2}) + (1-\beta) \cdot \sum_{k=1, k\neq j}^{n} \delta^{Ie}_{j|k}(v_j^{i_1}, v_k^{i_2})], \end{aligned} \tag{8}$$

where $\beta \in [0, 1]$ is the parameter that decides the weight of intra-coupled similarity, $v_j^{i_1}$ and $v_j^{i_2}$ are the values of feature j for instances u_{i_1} and u_{i_2}, respectively. δ_j^{Ia} and $\delta^{Ie}_{j|k}$ are the intra-coupled feature value similarity and inter-coupled feature value similarity, respectively. It is remarkable to note

that α_j is the feature weight defined in Equation (5), rather than $\alpha_j = 1/n$ assumed in [13].

E. Integration

Finally, we aggregate the membership assignment, feature weighting and similarity calculation, and propose an *Integrated Similarity* for classifying class-imbalanced mixed type data sets.

The **Integrated Similarity** represents the adapted coupled similarity measure by taking into account the feature weight, feature values' intra and features inter coupled relationship as well as the class size information. Formally,

$$IS(u_e, u_i) = \theta(C(u_i)) \cdot AS(u_e, u_i), \tag{9}$$

where u_e and u_i are the instances, respectively; $C(u_i)$ denotes the class of u_i; $\theta(\cdot)$ is the sized membership of class defined in Equation (3); and $AS(\cdot)$ is the adapted coupled object similarity defined in Equation (8).

As indicated by Equation (9), on one hand, although we only choose two classes in our experiments, the $\theta(\cdot)$ can capture the class size information, which is the key clue to the class imbalance, so it can extends to the classification tasks with multiple classes. On the other hand, the adapted similarity $AS(\cdot)$ includes not only the feature-class coupling information (feature weight), but it also capture the feature values' intra-coupling relationship and values from different features' inter-coupling relationship. By doing data discretization, we break out the limit which coupled relationship can only be applied in categorical data set, and extend such strategy to mixed data type. Therefore, the similarity in our algorithm is more reasonable than that in the existing similarity calculation related algorithms for the imbalanced real world mixed type data.

In this work, we illustrate our method by kNN. After obtaining the similarity between the instances u_e and $\{u_i\}$, we choose the k nearest neighbors that correspond to the k highest similarity values. The most frequently occurred class c_f in the k neighbors is the desired class for u_e. For example, in Table I, we have $IS(u_0, u_1) = 3.9785$, $IS(u_0, u_2) = 3.8054$ and $IS(u_0, u_5) = 3.8332$ to be the top three nearest neighbors to u_0, so u_0 should be labeled as its real class, class A (with $k = 3$).

V. Experiments and Evaluation

A. Experiments Setting

As the publicly available data sets were often not designed for the non-IIDness test as in this work, we choose the commonly used UCI and KEEL data and some real world data, which are all contain both numerical and categorical features. Our motivation is that if an algorithm can show improvement on such data compared to the baselines, it has potential to differentiate itself from others in more complex data with strong couplings. In total, 10 data sets are taken from the UCI Data Repository [18], KEEL data set repository [19], and the real Student learning data taken from the records of an Australian university's students performance database (If a student failed both in course L and course S, he or she will be labeled as "Failure", or else be labeled as "Success"). A short description of all the datasets is provided in Table III and the imbalanced rate of the class is shown as *Minority(%)*. These data sets have been selected as they typically have an imbalance class distribution (the lowest one is 0.98%) and all contain both categorical and numerical features(as shown in III, the "#(N+C) Features" denotes the feature type and numbers). The data sets such as D9 and D10 which has a more balance class distribution are selected to evaluate our algorithm's expansion capability.

We conducted 10-fold cross validation experiments to evaluate the performance of all the algorithms. In the experiments, we not only select several variants of kNN, such as the classic K Nearest Neighbors(kNN)[20], kENN[4], CCW-kNN[5] and SMOTE based kNN to compare with, but also the very popular classifiers C4.5 and NaiveBayes. To make algorithms more comparable, we further incorporate our coupled fuzzy method into some kNN algorithms (the new ones are with a prefix of $HC+$) to compare their results. In all our experiments, we set $k = 5$ to all those kNN-based classifiers, and the confidence levels for kENN is set to 0.1.

Due to the dominative effect of the majority class, the overall accuracy is not an appropriate evaluation measure for the performance of classifiers on imbalanced datasets, we use Receiver Operating Characteristic (ROC) curve and the Area Under the ROC Curve (AUC)[21] to evaluate the performance results. AUC indicates the overall classification performance, and the AUC of a perfect classifier equals to 1, a bad one less than 0.5, so a good classification algorithm will has a higher AUC.

B. Results and Analysis

Table IV shows the AUC results for our CF-kNN compared with the state of the art algorithms. The top two results are highlighted in bold. Compared with other approaches, our CF-kNN has the highest AUC result and outperforms others in most of the datasets, especially in datasets with high imbalance rate. Also, our proposed CF-kNN always outperforms classic kNN on all the datasets. This evidences that considering the coupling relationships between objects, features and feature values by treating the data as non-IID in computing similarity or distance captures the intrinsic data characteristics. Note that the SMOTE-based kNN does not always demonstrate significant improvement compared with kNN, sometimes even worse, such as in data set D2 and D10. It means that only using SMOTE on imbalanced data may not bring much improvement, but even some noise.

From the results we can see that when the imbalance rates are less than 6.2%, our method achieves a much better improvement (the least one is 3.36% and the highest one is 12.09%) on these very simple UCI data which does not incorporate much non-IIDness characteristics. That confirms again that our coupled fuzzy strategy is very effective for imbalanced non-IID classification tasks.

Experiment 2 aims to test the effect of incorporating fuzzy membership of class and the coupled similarity into

TABLE III

DATA SETS, ORDERED IN THE DECREASING LEVEL OF IMBALANCE

Index	**Dataset**	**Source**	**#Instances**	**#(N+C) Features**	**#Class**	**Minority Name**	**Minority(%)**
D1	Student	REAL	50000	(24+8)	2	Failure	0.98%
D2	Abalone	UCI	4177	(7+1)	29	Class15	2.47%
D3	Annealing	UCI	798	(6+32)	5	U	4.26%
D4	Dermatology	UCI	366	(1+33)	6	P.R.P.	5.46%
D5	Census-Income	UCI	299285	(12+28)	2	5000+	6.20%
D6	Zoo	UCI	101	(1+16)	7	Set6	7.92%
D7	Contraceptive	UCI	1473	(2+7)	3	Long-term	22.61%
D8	Adult	UCI	45222	(6+8)	2	>50K	23.93%
D9	German Credit	KEEL	1000	(7+13)	2	bad	30.00%
D10	Credit Approval	UCI	690	(6+9)	2	positive	44.50%

TABLE IV

THE AUC RESULTS FOR CF-kNN IN COMPARISON WITH OTHER ALGORITHMS

Dataset	Minority(%)	**CF-kNN**	kNN	kENN	CCWkNN	SMOTE	C4.5	Naive	**improvement**
D1	0.98%	**0.909**	0.845	0.849	0.854	**0.866**	0.857	0.857	4.97%-7.59%
D2	2.47%	**0.718**	0.672	0.680	**0.692**	0.688	0.683	0.682	3.75%-6.89%
D3	4.26%	**0.768**	0.714	0.735	**0.743**	0.732	0.737	0.729	3.36%-7.49%
D4	5.46%	**0.76**	0.715	0.720	**0.729**	0.678	0.716	0.724	4.28%-12.09%
D5	6.20%	**0.815**	0.782	**0.803**	0.798	0.788	**0.803**	0.791	1.49%-4.28%
D6	7.92%	**0.887**	0.842	**0.869**	**0.869**	0.854	0.857	0.859	2.08%-5.30%
D7	22.61%	**0.755**	0.718	0.729	0.725	**0.743**	0.726	0.736	1.64%-5.12%
D8	23.93%	**0.938**	0.904	0.915	0.910	0.910	**0.920**	0.919	1.95%-3.79%
D9	29.72%	**0.769**	0.738	**0.757**	0.744	0.755	0.752	0.756	1.53%-4.24%
D10	44.50%	**0.916**	0.893	**0.913**	0.910	0.887	0.907	0.912	0.33%-3.27%

other classification algorithms. For doing this, we create three comparison sets by integrating the proposed coupled fuzzy mechanism into kENN to form CF+kENN, CCWkNN to form CF+CCWkNN, and SMOTE based kNN to form CF+SMOTE based kNN, and compare their performance. All comparable algorithms are with the same parameter settings.

Table V shows the performance results of these comparable algorithms with vs. without the coupled fuzzy mechanism. It shows that incorporating our new similarity metrics will bring more or less improvement for the classic algorithms, especially for those distance or similarity-based algorithms. This further shows that our proposed idea of incorporating the fuzzy membership of classes size and measuring the couplings between objects, features and feature values capture the intrinsic characteristics better than existing methods, and it especially suitable for class-imbalanced data.

To evaluate our coupled similarity on different imbalance rate, we do SMOTE on student data and create 50 new data sets, in which the minority class varies from 1% to 50% of the total instances. Fig. 1 shows the improvement of the basic algorithms which combined with our Coupled Fuzzy Similarity on different imbalance rate. As it shows in the figure, when minority class only takes up $< 10\%$ of the total instances, both kNN and kENN (combined with CF) can have an improvement of over 5.821%. Even for CCWkNN, the improvement can over 5.372%. But with the imbalance rate declining, this improvement falls simultaneously. When minority class comes to 35% of the total records (which can be defined as "balanced" data) or over, the improvement will not be so outstanding and stay stable at about 2.2%. This experiment demonstrates that our strategy is sensitive to the imbalance rate, and it is more suitable for being used in the scenario with high imbalance rate, that is, imbalanced mixed type Non-IID data.

VI. CONCLUSION AND FUTURE WORK

Traditional classifiers mainly focus on dealing with balanced data set and overlook the couplings between data attributes, objects and classes. Classifying coupled and imbalanced data is very challenging. We propose a hybrid coupled kNN to partition imbalanced mixed type data with strong relationships between objects, attributes and classes. It incorporates the sized membership of a class with feature weight into a coupled similarity measure, which effectively extracts the inter and intra coupling relationships between feature values. The experiment results show that our HC-kNN has a more stable and higher average performance than the regular kNN, kENN, CCWkNN, SMOTE-based kNN, Decision Tree and NaiveBayes when applied for class-imbalanced mixed

TABLE V

THE AUC RESULT COMPARISON FOR ALGORITHMS WITH AND WITHOUT COUPLED FUZZY METHOD

Dataset	Minority(%)	kENN	CF+kENN	CCWkNN	CF+CCWkNN	SMOTE	CF+SMOTE
D1	0.98%	0.849	**0.905**	0.854	**0.906**	0.866	**0.922**
D2	2.47%	0.680	**0.724**	0.692	**0.733**	0.688	**0.735**
D3	4.26%	0.735	**0.783**	0.743	**0.788**	0.732	**0.778**
D4	5.46%	0.720	**0.766**	0.729	**0.771**	0.678	**0.718**
D5	6.20%	0.803	**0.912**	0.798	**0.873**	0.788	**0.836**
D6	7.92%	0.869	**0.922**	0.869	**0.918**	0.854	**0.908**
D7	22.61%	0.729	**0.764**	0.725	**0.725**	0.743	**0.776**
D8	23.93%	0.915	**0.957**	0.910	**0.946**	0.910	**0.951**
D9	30.00%	0.757	**0.780**	0.744	**0.785**	0.755	**0.800**
D10	44.50%	0.913	**0.936**	0.910	**0.932**	0.887	**0.907**

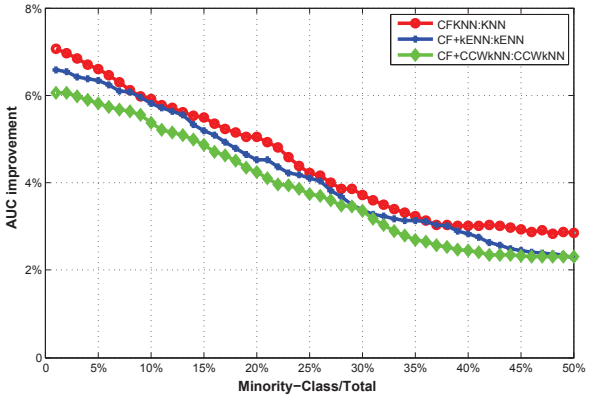

Fig. 1. The sensitivity to imbalance rate.

type data. Future work will include increasing the algorithm efficiency and lowering the time complexity, and applying this idea to other basic classification algorithms based on similarity or distance.

REFERENCES

[1] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. Wiley-IEEE Press, 2011.

[2] T. Pang-Ning, M. Steinbach, V. Kumar *et al.*, "Introduction to data mining," in *Library of Congress*, 2006, p. 74.

[3] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *SDM 2008*, 2008, pp. 243–254.

[4] L. Yuxuan and X. Zhang, "Improving k nearest neighbor with exemplar generalization for imbalanced classification," in *15th Pacific-Asia Conference, PAKDD 2011*. Springer, 2011, pp. 1–12.

[5] W. Liu and S. Chawla, "Class confidence weighted knn algorithms for imbalanced data sets," *Advances in Knowledge Discovery and Data Mining*, pp. 345–356, 2011.

[6] Q. Yang and X. Wu, "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol. 5, no. 4, pp. 597–604, 2006.

[7] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[8] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[9] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[10] W. Liu, S. Chawla, D. Cieslak, and N. Chawla, "A robust decision tree algorithm for imbalanced data sets," in *SDM 2010*, 2010, pp. 766–777.

[11] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 435–442.

[12] Y. Song, J. Huang, D. Zhou, H. Zha, and C. L. Giles, "Iknn: Informative k-nearest neighbor pattern classification," in *Knowledge Discovery in Databases: PKDD 2007*. Springer, 2007, pp. 248–264.

[13] T. Yang, L. Cao, and C. Zhang, "A novel prototype reduction method for the k-nearest neighbor algorithm with $k \geq 1$," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2010, pp. 89–100.

[14] C. Wang, L. Cao, M. Wang, J. Li, W. Wei, and Y. Ou, "Coupled nominal similarity in unsupervised learning," in *CIKM 2011*. ACM, 2011, pp. 973–978.

[15] L. A. Kurgan and K. J. Cios, "CAIM discretization algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 2, pp. 145–153, 2004.

[16] T. J. Ross, *Fuzzy Logic with Engineering Applications*. John Wiley & Sons, 2009.

[17] A. Ahmad and L. Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data and Knowledge Engineering*, vol. 63, no. 2, pp. 503–527, 2007.

[18] K. Bache and M. Lichman, "UCI machine learning repository," 2013.

[19] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2010.

[20] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.

[21] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.