2012 4th Conference on Data Mining and Optimization (DMO)
02-04 September 2012, Langkawi, Malaysia

A Direct Ensemble Classifier for Imbalanced Multiclass Learning

Mohd Shamrie Sainin
School of Computing, College of Arts and Sciences,
Universiti Utara Malaysia
Sintok, Kedah, Malaysia.
shamrie@uum.edu.my

Rayner Alfred
School of Engineering and Information Technology
Universiti Malaysia Sabah, Jalan UMS,
Kota Kinabalu, Sabah, Malaysia.
ralfred@ums.edu.my

***Abstract*—Researchers have shown that although traditional direct classifier algorithm can be easily applied to multiclass classification, the performance of a single classifier is decreased with the existence of imbalance data in multiclass classification tasks. Thus, ensemble of classifiers has emerged as one of the hot topics in multiclass classification tasks for imbalance problem for data mining and machine learning domain. Ensemble learning is an effective technique that has increasingly been adopted to combine multiple learning algorithms to improve overall prediction accuraciesand may outperform any single sophisticated classifiers. In this paper, an ensemble learner called a Direct Ensemble Classifier for Imbalanced Multiclass Learning (DECIML) that combines simple nearest neighbour and Naive Bayes algorithms is proposed. A combiner method called OR-tree is used to combine the decisions obtained from the ensemble classifiers. The DECIML framework has been tested with several benchmark dataset and shows promising results.**

Keywords-machine learning; data mining;data mining optimization; nearest neighbour;naive bayes; ensemble; classification;imbalance; multiclass

I. INTRODUCTION

A multiclass classification is a special case within statistical classification of assigning one of several class labels to an input object.Unlike the binary classification, learning multiclass problems is a more complex task to exploit as each example can only be assigned to more than two class labels. Most researches in classification tasks focus on binary classification. However classifiers that are designed for binary classification are not effective to be used in multiclass classification tasks [1]. Data with multiclass labels has more than two classes. An imbalance data with multiclass labels refers to a dataset with target class which is skewed in distribution. With the existence of imbalance data in a multiclass classification task, traditional classification methods cannot be applied efficiently and effectively since they generally assume data are well distributed [2].

Generally, there are three categories of method proposed for multiclass classification tasks. They are 1) A direct multiclass classification technique using a single classifier (e.g., decision trees, neural networks, k-nearest neighbour, Naïve Bayes, and support vector machines); 2) A binary classification conversion; 3) A hierarchical classification.

A direct classifier is any algorithms which can be applied to a classification problem directly and they are naturally extensible from a binary classification task. These algorithms include neural networks, decision trees, k-Nearest Neighbour, naïve Bayes and Support Vector Machine [1]. In contrast, indirect methods that are applied to solve multiclass problem require steps to change or pre-process data into binary problem before any classification processes can be made to target class.

Researchers have shown that although a direct classifier algorithm can be easily applied to a multiclass classification, the performance of a single classifier is not efficient when applied to classification of multiclass problemsfor imbalance data. Therefore, an ensemble of classifiers has emerged as one of the hot topics in learning imbalance data with multiclass labels recently [1].

Thus, in this paper, an imbalanced multiclass classification problem is investigated and a method to solve the problem based on the proposed ensemble of two different classifiers, a naïve Bayes and a Nearest Neighbour technique, is described in this paper. Specifically, the proposed method is utilizing the simple instance based learning (k-nearest neighbour) and the probability based learning (naïve Bayes) with certain modification such as classification decision combiner, in classifying a multiclass problem with imbalance data.In this paper, we are particularly interested in learning a multiclass classification task for imbalance data due to several important reasons; 1) There have been many attempts to deal with class imbalance, yet many of these studies focus on binary classification which shown to be less effectivein multiclass classification [1], 2) Imbalance situation is even more complicated for multiclass classification, as more attention is required to handle the imbalance between multiple pattern classes [3], and 3) In practice, most applications have more than two classes where imbalance distributions hinder the classification performance.

II. BACKGROUND

The general classification problem known in data mining and machine learning domain is the problem of mapping an observed feature vector into a predefined class. There are two types of classification which are binary and multiclass classifications. A binary classification involves only two classes, whereas there are more than two classesdefined in a

978-1-4673-2718-3/12/$31.00 ©2012 IEEE

multiclass classification. While two-classes or binary classification problem is already well formulated [2], a multiclass classification problem is still receiving much attention due to its wide applications in real world data [4]. A multiclass classification problem is defined as the task of assigning a class label to an object which has more than two predefined classes.

Multiclass classification problems are very much required in real world applications for the tasks like object recognition, character recognition, person recognition, disease diagnosis and many more. Example of real world applications of multiclass methods can be seen in these recent literatures [5, 6].

Direct single classifier algorithms are traditional classifiers which naturally extensible algorithms from binary classification technique. Many of the earlier construction of machine learning algorithms are considered as single classifier which was proposed for solving the binary and multiclass data classification [2]. Some of the learning algorithms have been broadly and theoretically studied for their effectiveness in various application domains that they become standard machine learning topics. Popular standard single classification methods are (NB) naïve Bayes [7] and k-nearest neighbour (kNN) algorithm [8], artificial neural network, decision tree, and support vector machines [9]. Apart from successfully applied to the binary classification problem, these algorithms can also applied directly to multiclass classification technique [1], where they can be applied directly to the multiclass classification problem without heavy modification of the algorithms or the data. However, some of the direct approaches combine pattern modelling schemes such as one-against-one (OAO), one-against-all (OAA) and P-against-Q (PAQ) in order to tackle the multiclass classification problem.

Naïve Bayes (NB) is one of the practical Bayesian learning methods and also often called as the naïve Bayes Classifier [10, 11]. It is based on a principle of probabilistic modeling based on (MAP) Maximum A Posteriori principle [11] that incorporate strong independence assumptions that has no bearing in reality, hence called as 'naïve' [12]. Naïve Bayes is shown to be effective on applications where there are many probabilities need to be computed such as text categorization [13] and spam filtering [14]. Text categorization and spam filtering share the similar identities, where the amount of attribute may extensive in size and also a multiclass relational problem.

The k-nearest neighbor (kNN) is one of the simplest and oldest nonparametric classification algorithm which first introduced by Cover and Hart[8], which uses distance measure technique. Basically, the algorithm is designed to work as instance based learning by assuming all instances can be measured using distances represented in n-dimensional space [10]. The entire training samples are stored in computer memory record (database), thus no global model is created other than local estimations (distance) on future unseen instances. Then, the algorithm finds k examples in the training sample that are closest to new test instance [12]. In other words, the distances of the new test instance to each instance in training sample are computed and specify the k-nearest to the test instance.The nearest neighbor algorithm is popular due to its advantages [15], 1) Conceptual simplicity, however able to use more complex, symbolic representation for instances [10], 2) Easy implementation: Training is just storing all samples and classification is using common distance measure such as Euclidean distance, 3) Known error rates bounds (explained in detail by [8]), and 4) Comparable to other strong classifier in real applications: Under mild condition, the kNN rule able to perform competitively even with the large sample size [16]. The kNNwas also applied to multiclass problems [17] and multiclass with imbalance problem [18].

Although that kNN learning is criticized for its drawbacks, however many approaches have been proposed to deal with the problems of kNN. Additional proposed methods to address the drawbacks of nearest neighbor learning can be referred in [19]. More importantly, the learning algorithm is normally being used as a benchmark in various classification studies [1].

Through the recent development in data mining research, imbalance data has emerged as one of the most important issues arises from the rapid contributions of academic research to the real world application (i.e. applied science). Imbalance data has also been identified as one of the most challenging problems in machine learning and data mining due to its significant effects to classifier performance [20]. Solutions for bi-class problems are not applicable directly to multiclass cases. Possible solution such as multiclass conversion to several bi-class (i.e. OVA); yet the obvious drawback are 1) to learn an identification model for each class label assignment is expensive in training and 2) decision can be made differently on every comparisons, and 3) one class versus the other classes will worse in imbalanced distribution [21].

Imbalance or also known as skewed data is a problem exists in a sample data when certain class is represented by a significantly small number of examples compared to other classes [22]. The problem of imbalance can be determined by two components which are distinguished by the ratio and lack of information [23]. Ratio is first mentioned in [23] which defines the imbalance ratio (IR) as follows:

$$IR = \frac{NumberOfMinority}{NumberOfMajority}$$

where, $NumberOfMinority$ is class with very few instances and $NumberOfMajority$ is class with significantly large number of instances. Meanwhile, the lack of information (LI) is the problem of very few information for the minority class.

Furthermore, [24] stated that imbalance can be measured by the ratio of the size of the training data in the smallest class vs. the largest class. Suppose that in K-class classification, n_i denote the number of data examples in class i in training data D, the imbalance measure of D is:

$$\beta_D = \frac{min\{n_i |\ i = 1, \dots, K\}}{max\{n_i |\ i = 1, \dots, K\}}$$

There is no benchmark ratio which is used as a standard measure in any imbalance learning other than the description discussed in three research papers above. Recently, Ding proposed a threshold to define the imbalance learning problem for data mining community[2]. Imbalance ratio (based on 2-class problem) is defined as the ratio measured from the majority class and minority class as follows:

$$\beta = N^{-}:N^{+}$$

whereN^{-} is the size of negative class, N^{+} is the size of positive class and β can always bigger than 1. Thus, the bigger the value of β, the more skewed the data. According to Ding, a learning problem is a significantly imbalanced classification problem (or simply imbalance learning) if ratio is no less than 19:1 or the size of minority class is only 5% of the entire sample data, for general binary classification [2]. The proposed ratio is based on the ideas that 1) the threshold can be used in multiclass imbalance problem, 2) the statistical testing define 5% significance level and 3) the threshold only limits the scope of theoretical studies on imbalance learning, while the comparison can still be made to the less significant or moderate imbalanced data.

It is commonly agreed in many research papers on imbalanced problem stated that due to the class imbalance, the performance of a learning algorithm is degraded and the results obtained are favoring the majority class. This is due to several reasons that include, 1) accuracy driven (minimize error), while the minority class contributes significantly low, 2) equal distribution of data assumption and 3) assuming that errors from different class have the same cost [25].Methods for imbalance problems can be addressed infour approaches, namely sampling, algorithms, ensemble, and feature selection.

An ensemble classifier is defined as a classifier that consists of individual trained classifier that applies many similar single classifiers or combines two or more different classifiers whose decisions will be combined when classifying new unseen instances. It is theoretically and empirically studied that ensembles are not only more accurate than the single predictive models, but they are also very diverse in learning data. The reason of why ensemble methods are able to outperform any single classifiers were discussed in [25]. He has shown that boosting-based ensembles using decision tree classifiers (C4.5) indicate that all the findingsobtained by [26] are true, in which ensembles will always outperform single classifiers due to the improvements on the three areas, namely the statistical problem, the computational problem and the representation problem. A single classifier suffers mostly in these three areas due to the fact that a single hypothesis is largely depending on training data (if several other hypotheses give the same accuracy – statistical problem), the best hypothesis is not guaranteed (computational problem) and the hypothesis may not available in the search space (representational problem).

There are four approaches used in the construction of ensemble classifiers, which are, 1) Different combination schemes (combination level) – addressing the problem to pick a combination scheme and train it of necessary (This approach will find the best combination of ensembles of similar training data), 2) Different classifier models (classifier level), 3) Different feature subset (feature level), and 4) Different training set (data level) [25]. Two popular combination of classifiers prediction methods are algebraic combiners and voting based methods [27]. Algebraic combiners are non-trainable and usually produce continuous valued outputs. They can be used in combining the prediction by using the algebraic expression such as minimum, maximum, sum, mean, product, median, and etc. Final ensemble decision is calculated based on the largest support after the certain algebraic expression is applied, thus assigning the class label to the new instance. Methods based on voting are simple and widely used for classifiers prediction combiner [28]. Voting methods works by using labels only where if c_j gets the largest voting total, then the ensemble will choose the class label for classifying an unseen data. When there is no majority vote, a coil state (a common state) is used. There are two types of voting schemes used ([27], simple majority voting and weighted majority voting.

Other combination rules are linear discrimination, neural networks and decision trees [28]. However these combination methods depend on the algorithms and they are less used in the literatures. Combining methods discussed here are all estimate a class label outputted from the ensemble classifier. In the multiclass classification case, decision can be estimated not only by single class but other appropriate class. Interestingly, many real-world multiclass classification problems can be represented into a setting where non-crisp label needs to be observed such as scene classification in computer vision, literature categorization, social network analysis, agriculture (crop-land analysis), microbiology and etc. [29]. Non-crisp label is the specification of class label into degree of membership which normally used in fuzzy classifiers.

Another intuitive similar idea is explained in [30]. They proposed the direct estimation of class membership probabilities for multiclass classification using multiple scores. The idea behind the approach is to take the advantage of multiclass classification and multiple scores are computed for all of the classes without decomposition. The reason for this method is that the predicted class is determined not by the absolute value of the score but by using the relative position among the score. Therefore, the class membership in their method not only depends on the score for the target class but also on other scores.

The possibility of alternative prediction combiner in the ensemble for multiclass classification problem can be formulated [31, 32]. Instead of giving one label in the final classification, two class labels with high weighted voting represented as probabilities are combined as an OR-tree combiner. OR-tree combiner works by not only providing a pool of decision probabilities but by combining the strength of selected ensembles classifiers (on both agreement and disagreement of the classifiers).

Motivated by the drawbacks and possible gain through single classifier algorithm, ensembles method is investigated to solve the imbalanced multiclass classification problem in different perspectives. The ensemble of relatively weak (but better than random guessing algorithms) single classifier algorithms are studied for their contribution in the problem. Particular algorithms that are focused in this paper are the naïve Bayes and k-nearest neighbor algorithms, where the ensemble consists of either single classifier or by combining both classifiers to create diverse ensemble.

The literature survey on the combination of NB and kNN shows that there is only few works that devoted in this combination for the ensembles to specifically addressing the imbalanced multiclass classification problem. Among nearest examples we can find on this specific problem are [31-33]. Work by [12] is perhaps the most current ensemble that utilizing the combination of NB and kNN. However, additional ID3 classifiers introduce the complexity of the ensembles to certain degree and thus can only be used to nominal type of data.

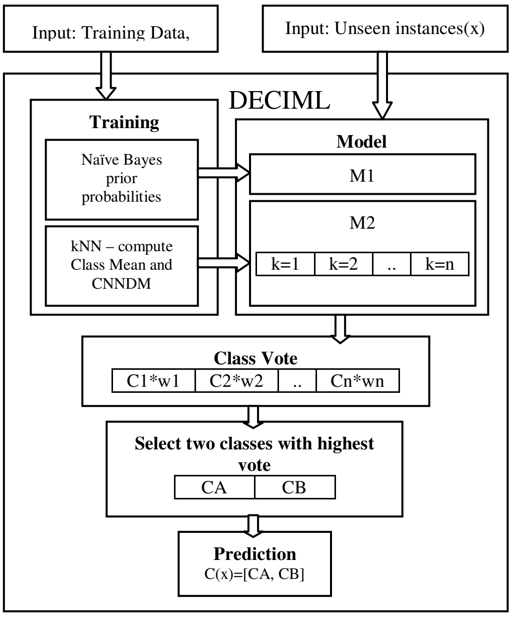
III. THE DECIML FRAMEWORK

The DECIML algorithm is proposed by combining two direct single classifiers that can be used for learning imbalancedmulticlass data(with no modification and with Bootstrap Aggregation). The proposed DECIML also utilizes the OR-tree prediction combiner. The aim for DECIML is to build a straightforward ensemble algorithm which works for multiclass imbalance learning as described in Figure 1 and the general flow of the algorithm is depicted in Figure 2.

```
Algorithm: DECIML
Inputs:Learner - base learning algorithm (NB as
M1 and 1NN or kNN as M2)
D - set of m training examples
T - set of testing examples
Outputs:
M - a composite model of M1 and M2
Learning Procedure:
1.  If learning is using default data then use D
2.  Else, generate randomly new training D′ with
    equal number of examples from D using
    selection with replacement technique.
    Instances that did not make it into D′ will
    form a testing set, T′
3.  Learn NB and derive M1 (consists of prior
    probabilities of the training data)
4.  If k=1 for kNN, then learn 1NN and derive M2
5.  Else if k>1 for kNN, then learn kNN with
    voting majority and derive M2
6.  Classify D using M1 and M2; derive weight for
    M1 and M2 based on accuracy.
Classification procedure:
1.  Classify T or T′ using M1, derive vote M1v
2.  Classify T or T′ using M2, derive vote M2v
3.  Combine  M1v and M2v using OR rule prediction
    combiner
4.  Returns c(x) = {hM1(x) OR hM2(x)}
```

Figure 1. The algorithm for DECIML

Based on Figure 1, first, the ensemble is initialized with two base classifiers (NB+1NN or NB+kNN). Given a benchmark data, D, of m examples with A_n numeric-valued attributes $\{ A_1, A_2, \dots, A_n \}$ and set of multilabel class $C = \{C_1, C_2, \dots, C_n\}$, each instance is represented by $X_m =< x_1, x_2, \dots, x_n, C(x) >$, where x_n is the numeric value of attribute A_n and $C(x)$ is the class label from C. In the implementation of DECIML, several types of data preparations will be used and observed, which includes using the 60:40 splitting or using default training and test from the repositories, 5-cross validation dataset, and/or selection with replacement technique.



Input: Training Data,
Input: Unseen instances(x)
DECIML
Training
Naïve Bayes prior probabilities
kNN – compute Class Mean and CNNDM
Model
M1
M2
k=1
k=2
..
k=n
Class Vote
C1*w1
C2*w2
..
Cn*wn
Select two classes with highest vote
CA
CB
Prediction
C(x)=[CA, CB]

Figure 2. General flow of DECIML framework

In order to train the DECIML, each direct single classifier will be trained using *D* or *D'* to derive a learning model *M*. NB learning model consists of probabilistic estimations (class-conditional probabilities) that incorporate strong independence assumptions [12] of a given dataset *D*. Although that 1NN or kNN will not generalize to a global model of D other than local model of (all instances in the training), however the class center (mean) nearest neighbor distance matrix (CNNDM) of each *C* will be calculated to support the voting later in prediction. The similar work of NNDM has been discussed in [34]. Figure 3 describes the algorithm to determine the class centers similarity matrix.

```
Algorithm: CNNDM
Inputs: D - Set of m training examples, μ -
Weighting parameter, μ=0.5
Outputs:
CNNDM- class center distance matrix
Procedure:
For i=0 to Cn
    a.  For instance m=0 to xm in D
    b.  Calculate distances of x to class mean Ci
    c.  Dist[m]=distance(m, Mean(Ci))
    d.  Index[m]=m
    e.  Calculate pull[m]
    f.  End For
    g.  Sort Index based on Dist
    h.  Create CNNDM[i][m] = Index
    i.  Apply loss function to CNNDM
          i.  For j=0 to n in CNNDM
         ii.  Calculate push for instance in
              CNNDM[i][j] = push[j]
        iii.  Loss = ((1-u)*pull[j]) +
              (μ*push[j])
         iv.  Update distance
          v.  End for
    j.  Update CNNDM
Return final CNNDM
```

Figure 3. Class center nearest neighbor distance matrix (CNNDM) algorithm

Class center Nearest Neighbor Distance Matrix or simply CNNDM is a similarity matrix of class center C_n to instances from training data D. Consequently, it will support the kNN voting in DECIML. Next, in the classification phase both classifiers in DECIML will predict their class value and each vote will be accumulated in class vote. Furthermore, prediction combiner will take place to combine the vote of the classifiers model (*M1* and *M2*). As mentioned before, the combiner used for ensemble method in the study is called OR-tree rule prediction combiner. This method works by examining two highest votes among n classes, so that it provides two prediction outputs in the form of$c(x) = \{hM1(x) OR hM2(x)\}$. It provides more flexible classification of multiclass whereby $c(x)$can be represented as $[A = hM1(x), B = hM2(x)]$. A is the prediction value with highest or at least similar weighted vote with B, while A and B have the highest weighted vote than other class values in $C(x)$. Thus, instance will be classified based on the rule $c(x) = \{AORB\}$ and this is only valid with multiclass data classification problem. Figure 4 shows the detailed algorithm using OR prediction combiner.

IV. Experimental Design and Results

This section examines and verifies the performance of the proposed ensemble method on several imbalanced data sets. Comparisons of performance were carried out among several methods for the direct learning algorithm in multiclass imbalance learning problem. As described earlier, the NB and kNN are combined in DECIML and used as direct algorithms in learning multiclass data. Furthermore, ensemble methods found in Weka specifically Bagging and AdaBoostM1 together with popular base learners were used in the experiments for comparison. The results obtained indicate that the proposed DECIML framework is able to perform on various types of imbalanced data and other benchmark dataset.

In this study, three base learners particularly the decision stump (DS), decision tree (DT), and multilayer perceptron neural network (MLP) are considered in the experiments with two ensemble frameworks, namely, AdaboostM1 and Bagging. The base learner in DECIML will use two algorithms namely naïve Bayes (NB) and nearest neighbor algorithm (NN). There are two ensembles in DECIML framework that consists of NB+1NN (naïve Bayed and 1-nearest neighbor) and NB+kNN (naive Bayes and k-nearest neighbor). Therefore the total number of ensemble learning algorithms to be evaluated is 2*3 (AdaBoostM1 and Bagging) + 1 (Random Forest) + 2 (NB+1NN and NB+kNN) = 9 in five group of ensembles (AdaBoostM1, Bagging, Random Forest, DECIML-NB+1NN and DECIML-NB+kNN).

In order to create the benchmark pool, the publicly available dataset repositories were examined such as the UCI [35], KEEL [36], UCR Time Series [37], NIPS Feature Selection Challenge [38] and previously used dataset in multiclass imbalance publication (IEEE, ACM, Springer, Science Direct, etc). There are 16 selected datasets were chosen for multiclass imbalance data in 5 different domains, example size vary from 100 to 50,000, feature size change from less than 10 to 100 and imbalance ratio range from 1:2 to 1:4559. The detailed properties of the benchmark dataset in this paper are carefully summarized in Table V.

```
Algorithm 3: OR-tree Rule Prediction Algorithm
Inputs:
V- Votes vector from NB(M1) and kNN(M2)
x- unseen instance vector
β- distance threshold from class mean
Outputs:
C(x)=[A,B]- OR classification of x; $c(x) =$
$\{A = hM1(x) OR B = hM2(x)\}$

Procedure:
1. Determine max vote from V, assign class with
   max vote, cM1
2. Determine combiner rule
    3. If  $\boldsymbol{cM1(x) = hM1(x)}$ and $\boldsymbol{cM1(x) = hM2(x)}$
          a. d1 = distance(Mean($\boldsymbol{cM1(x)}$), x)
          b. For i=0 to Mean(Ci)
                i. d2 = distance(Mean(Ci), x)
               ii. determine minimum d2 and
                   its class, cM2
          c. End for
          d. If d1 != d2min then
             $\boldsymbol{c(x) = \{cM1\ ORcM2\}}$
          e. Else
              i. Get class with max weight as
                 WC
             ii. Assign cM3 = WC
            iii. Then $\boldsymbol{c(x) = \{cM3\ ORcM1\}}$
  4. Else If  $\boldsymbol{cM1(x) != hM1(x)}$ and $\boldsymbol{cM1(x) !=}$
     $\boldsymbol{hM2(x)}$
       a. d1 = distance(Mean(hM1), x)
       b. d2 = distance(Mean(hM2), x)
       c. if β-d2 > β-d1 then cM4 = hM1
       d. else cM5 = hM2
       e. For i=0 to n in CNNDM
           i. classVote[Instance.Class(cM4)]++
          ii. classVote[Instance.Class(cM5)]++
       f. End for
       g. Determine weight(cM4) =
          cM4*classVote[cM4]
       h. Determine weight(cM5) =
          cM5*classVote[cM5]
       i. If cM4 = hM1 then $\boldsymbol{c(x) = \{cM4\ ORcM5\}}$
       j. Else if cM4 = hM2 then $\boldsymbol{c(x) = \{A =}$
          $\boldsymbol{cM5\ ORB = cM4\}}$
```

Figure 4. OR-tree rule prediction algorithm

In the experimental work, a comparison framework of several algorithms using a pool of benchmark dataset is performed. First, several single direct classifier performances are compared over the benchmark data, namely DS: Decision Stump; DT: Decision Tree; MLP: Multilayer Perceptron; NB: Naïve Bayes; 1NN: 1-Nearest Neighbor and kNN: k-Nearest Neighbor. The first framework is designed for the multiclass imbalance learning that includes Bagging, AdaBoostM1 and Random Forest. Random forest is one of direct imbalance learning algorithms, which is applied from Weka. Second, the experimental setup in DECIML is designed so that it applies naïve Bayes(NB), 1-Nearest Neighbor (1NN) and k-Nearest Neighbors (kNN) as the internal base classifiers. Note that, only two ensemble committee for DECIML based on NB and kNNin order to find the potentials of these two weak base classifiers. In order to prepare for the comparison, the first experiment framework is carried out by using a popular machine learning tool, Weka and the second framework consist of a proposed ensemble implementation of DECIML. After applying both experiment frameworks, their evaluation metrics (F-measure, G-means, MCC, and percentage performance) were carefully examined.

Comprehensive test was carried out using various settings in the internal procedure of the ensemble method. Basically, the DECIML consists of several steps for combining NB and 1NN or kNN ensembles classifiers, thus the general parameter setting will be used all experiments. The steps include priors for NB, loss function and nearest threshold setting for nearest neighbor distance matrix (NNDM) class centers. General values for loss function applied in this study is following the recommended settings by [39] where $\mu = 0.5$, $k = 5$ (target neighbors) and nearest threshold is set to $\tau = 10$ (for NNDM in DECIML).

Table I show the detailed performance metrics (using F-measure) of six single direct algorithms for multiclass imbalance on 16 benchmark dataset. Based on the results, we can easily examine that none of the algorithm performs significantly on every dataset. This indicates that the selected pool of benchmark dataset used in this study is fairly diverse and complicated. Also note that our target algorithms of NB, 1NN and kNN could produce almost similar performance with the strong algorithms such as DT and MLP. Next, Table II shows the performance metrics (F-measure only) using AdaBoostM1, Bagging and Random Forest ensemble methods. The results especially using DT and MLPNN as base classifier almost perform well on all dataset, thus this impose greater challenge for our ensemble method using direct algorithm combination in DECIML.

TABLE I. CLASSIFICATION PERFORMANCE (F-MEASURE) OF SIX DIRECT SINGLE ALGORITHMS (WEKA) ON 16 BENCHMARK DATASET.

Data	F-Measure					
	DS	DT	MLP	NB	1NN	kNN
Wine	0.569	0.949	0.972	0.966	0.955	0.955
Hayes-Roth	0.403	0.81	0.727	0.725	0.707	0.707
Contraceptive	0.256	0.510	0.523	0.494	0.434	0.476
Balance	0.563	0.761	0.91	0.86	0.776	0.875
Dermatology	0.350	0.962	0.976	0.97	0.957	0.957
Statlog (Landsat)	0.323	0.920	0.944	0.86	0.95	0.958
Glass	0.271	0.523	0.486	0.399	0.556	0.556
Car	0.530	0.890	0.854	0.714	0.871	0.858
Thyroid	0.915	0.992	0.947	0.298	0.922	0.922
New Thyroid	0.829	0.987	0.924	1.000	0.987	0.987
Nursery	0.567	0.872	0.909	0.874	0.861	0.94
Ecoli	0.511	0.820	0.844	0.849	0.807	0.862
Yeast	0.186	0.533	0.563	0.566	0.495	0.571
Pageblocks	0.912	0.970	0.956	0.921	0.959	0.959
Statlog (Shuttle)	0.858	0.999	0.997	0.908	0.999	0.999
Lympography	0.743	0.782	0.805	0.843	0.797	0.797
Average	**0.549**	**0.830**	**0.834**	**0.765**	**0.815**	**0.836**

TABLE II. CLASSIFICATION PERFORMANCE (F-MEASURE) OF ADABOOSTM1, BAGGING (USING THREE BASE CLASSIFIERS) AND RANDOM FOREST ON 16 BENCHMARK DATASET.

Data	AdaBoostM1 (F-Measure)			Bagging (F-Measure)			Random Forest (F-Measure)
	DS	DT	MLP	DS	DT	MLP	
Wine	0.91	0.97	0.97	0.84	0.93	0.98	0.98
Hayes-Roth	0.40	0.86	0.77	0.49	0.83	0.73	0.80
Contraceptive	0.26	0.52	0.53	0.26	0.52	0.56	0.50
Balance	0.70	0.80	0.92	0.64	0.80	0.92	0.81
Dermatology	0.35	0.95	0.98	0.35	0.97	0.97	0.96
Landsat	0.32	0.95	0.94	0.32	0.95	0.95	0.95
Glass	0.27	0.59	0.58	0.27	0.55	0.50	0.56
Car	0.53	0.90	0.90	0.53	0.82	0.90	0.90
Thyroid	0.97	0.99	0.94	0.92	0.99	0.94	0.99
New Thyroid	0.91	1.00	0.99	0.9	0.99	0.92	1.000
Nursery	0.57	0.88	0.93	0.57	0.87	0.92	0.88
Ecoli	0.51	0.86	0.83	0.51	0.84	0.88	0.84
Yeast	0.19	0.56	0.56	0.19	0.54	0.60	0.55
Pageblocks	0.91	0.97	0.96	0.91	0.97	0.96	0.97
Shuttle	0.99	1.00	0.99	0.86	0.99	0.99	1.00
Lympography	0.74	0.84	0.83	0.72	0.83	0.83	0.80
Average	**0.59**	**0.85**	**0.85**	**0.58**	**0.84**	**0.85**	**0.84**

The DECIML performance on the benchmark dataset is shown in Table III followed by Table IV which lists the average performance of all nine ensemble algorithms on the benchmark dataset using three evaluation metrics (F-measure, G-means and MCC). The values are the mean of metric values over 16 benchmark dataset with different splitting (training and testing) approach. As mentioned before, comparable ensemble method of AdaBoostM1 and Bagging are using their default settings as recommended in Weka and DECIML also with its components default parameter settings. However, instead of one base learner for other ensemble method, the DECIML is consists of two algorithms to specifically try to solve the multiclass imbalance problem directly. As we see from the metric average values, it shows that both combinations of algorithms in DECIML perform fairly good compared to its single algorithm only.

TABLE III. CLASSIFICATION PERFORMANCE (F-MEASURE) OF DECIML ON 16 BENCHMARK DATASET.

Data	F-Measure	
	NB+1NN	NB+kNN
Wine	1.000	1.000
Hayes-Roth	1.000	1.000
Contraceptive	0.814	0.845
Balance	0.824	0.870
Dermatology	0.978	0.978
Statlog (Landsat)	0.960	0.950
Glass	0.690	0.708
Car	0.806	0.806
Thyroid	0.920	0.930
New Thyroid	1.000	1.000
Nursery	0.966	0.960
Ecoli	0.685	0.703
Yeast	0.718	0.699
Pageblocks	0.980	0.980
Statlog (Shuttle)	0.902	0.945
Lympography	0.709	0.740
Average	**0.872**	**0.882**

Through the observation of the results obtained, an ensemble method using both AdaBoostM1 and Bagging algorithms almost produce similar results for all the datasets used, except for the ensemble method constructed with decision stump base learner. This shows that DT, MLP and Random Forest can be used as a base learner to multiclass imbalance problem. In addition to that, by combining two classifiers in an ensemble DECIML method, the performance of the prediction task can be improved where it slightly outperforms other methods in our experiments.

TABLE IV. OVERALL AVERAGE CLASSIFICATION PERFORMANCE (F-MEASURE, G-MEANS, MCC) OF 9 ENSEMBLE ALGORITHMS ON 16 BENCHMARK DATASET.

Ensemble	Base Learner	F-measure	G-Means	MCC
AdaBoostM1	DS	0.597	0.549	0.529
Bagging	DS	0.579	0.507	0.508
AdaBoostM1	DT	0.852	0.822	0.793
Bagging	DT	0.837	0.797	0.779
AdaBoostM1	MLP	0.851	0.779	0.797
Bagging	MLP	0.847	0.762	0.783
Random Forest		0.844	0.800	0.781
DECIML	**NB+1NN**	**0.872**	**0.882**	**0.904**
DECIML	**NB+kNN**	**0.882**	**0.897**	**0.923**

Close observation on the average results of the base learners, the DECIML framework produces fairly stable and a comparable performance throughout the benchmark dataset can be seen on the three evaluation metrics. Thus, it strongly supports the expectation where ensemble strategies (by combining two or more classifiers) are much more effective than individual learning algorithm approach for direct method in order to solve multiclass imbalance problem. Interestingly in these results, Random Forest is also consistent over the three evaluation metrics. This shows that the ensemble method constructed using several decision trees is worth to be combined with other strong classifiers for future observation. Although that the ensemble method using Bagging is ranked the last among the five methods, it still shows a similar trend with AdaBoostM1. Both ensemble method Bagging and AdaBoostM1 using decision stump are not effective for highly imbalance multiclass data.

In addition to that, note that the three metric evaluations are not consistent in measuring the classification performance due to their approach to represent algorithm accuracy over the data. F-measure is used to measure the true positive rate prediction and so as the accuracy of positive prediction, while G-means is used for the forecasting of the combined true positive and true negative. [2] stated that choosing suitable evaluation metric depends on the practical imbalance task, dataset and intention as follows: 1) if the precision and recall as main concern, then F-measure is the choice, 2) if the accuracies on positive and negative classes are important, then G-means a recommended, and 3) if no specific desire on accuracy of positive or negative class, then MCC is useful to produce general balanced classification on the overall performance.

V. CONCLUSION

This paper introduces an ensemble method applied for imbalanced multiclass learning, which is called Direct Ensemble Classifier for Imbalanced Multiclass Learning (DECIML). The method consists of two direct single algorithms that isspecifically used to address the imbalanced multiclass data problem with the Class Nearest Neighbour Distance Matrix (CNNDM) and OR-tree decision combination. Relatively extensive experiments have been done to 1) compare multiclass imbalanced classification performance between several ensemble strategies, 2) verify the effectiveness of the proposed DECIML ensemble that consists of NB+1NN and NB+kNN. Based on the results, the following is the summary of the observations: 1) It is widely accepted in machine learning domain that no single best algorithm or ensemble of algorithms for variety of data domain. Although that the performance of DECIML ensemble is top compared with other methods which observed in this study, each algorithm has their own strength and weakness in classifying different domain of data; 2) In line with [2] on their binary class imbalance problem evaluation, the three imbalanced evaluation metrics, the F-measure, G-means and MCC show similar trend. They are not consistent for evaluating the performance of a classifier. Therefore, one can use any evaluation metric depending on the purpose of the works; 3) The DECIML frameworks which consist of different ensemble of NB+1NN or NB+kNN are superior to individual classifier. However, several settings and implementation of the components in DECIML still need to be carefully tuned for future real world problem deployment.Future works related to the proposed ensemble will incorporate feature selection method to further improve the classification performance in imbalanced multiclass problem.

REFERENCES

[1] Lerteerawong, B. and M. Athimethphat, An Empirical Study of Multiclass Classification with Class Imbalance Problems, in International Conference on Business and Information, BAI2011. 2011: Sapporo, Japan.

[2] Ding, Z., Diversified Ensemble Classifiers for Highly Imbalanced Data Learning and Their Application in Bioinformatics, in Computer Science Department, Georgia State University. 2011, Georgia State University. p. Paper 60.

[3] Ghanem, A.S., S. Venkatesh, and G. West. Multi-class Pattern Classification in Imbalanced Data. in 20th International Conference on Pattern Recognition (ICPR), 2010 2010.

[4] Xia, Y. and Y. Ying. A Cooperative Learning Algorithm for Multiclass Classification. in Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. 2010.

[5] Escalera, S., et al., Multi-class Classification in Image Analysis via Error-Correcting Output Codes Innovations in Intelligent Image Analysis, H. Kwasnicka and L. Jain, Editors. 2011, Springer Berlin / Heidelberg. p. 7-29.

[6] Tapia, E., et al., Multiclass classification of microarray data samples with a reduced number of genes. BMC Bioinformatics, 2011. **12**(1): p. 59.

[7] Lowd, D. and P. Domingos, Naive Bayes models for probability estimation, in Proceedings of the 22nd international conference on Machine learning. 2005, ACM: Bonn, Germany. p. 529-536.

[8] Cover, T.M. and P.E. Hart, Nearest Neighbor Pattern Classification. IEEE Transactions in Information Theory, IT-13, 1967: p. 21-27.

[9] Aly, M. (2005) Survey on multiclass classification methods. pp. 1-9.

[10] Mitchell, T.M., Machine Learning. 1997: MIT Press and The McGraw-Hill Companies, Inc.

[11] Rish, I., An empirical study of the naive Bayes classifier, in IJCAI Workshop on Empirical Methods in Artificial Intelligence. 2001.

[12] Farid, D.M., M.Z. Rahman, and C.M. Rahman, An Ensemble Approach to Classifier Construction based on Bootstrap Aggregation. International Journal of Computer Applications (0975 – 8887), 2011. **25**(5): p. 30-34.

[13] Jiang, Y., et al., A technique for improving the performance of naive bayes text classification, in Proceedings of the 2011 international conference on Web information systems and mining - Volume Part II. 2011, Springer-Verlag: Taiyuan, China. p. 196-203.

[14] Qin, L., et al. Research of a Spam Filtering Algorithm Based on Naive Bayes and AIS. in Computational and Information Sciences (ICCIS), 2010 International Conference on. 2010.

[15] Barandela, R., R.M. Valdovinos, and J.S. Sánchez, New Applications of Ensembles of Classifiers. Pattern Analysis & Applications, 2003. **6**(3).

[16] Zhang, H., et al., SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. Proceedings of Conference on Computer Vision and Pattern Recognition, 2006: p. 2126-2136.

[17] Rokach, L., Ensemble-based classifiers. Artificial Intelligent Rev, 2010. **33**(1-2): p. 1–39.

[18] Liao, T.W., Classification of weld flaws with imbalanced class data. Expert Systems with Applications, 2008. **35**: p. 1041-1052.

[19] Anil, K.G., On optimum choice of k in nearest neighbor classification. Computational Statistics and Data Analysis, 2006. **50**: p. 3113-3123.

[20] Yang, Q. and X. Wu, 10 challenging problems in data mining research. International Journal of Information Technology and Decision Making, 2006. **5**(4): p. 597-604.

[21] Sun, Y., M.S. Kamel, and Y. Wang. Boosting for Learning Multiple Classes with Imbalanced Class Distribution. in Sixth International Conference on Data Mining, ICDM '06. 2006.

[22. Japkowicz, N. and S. Stephen, The class imbalance problem: A systematic study. Intell. Data Anal., 2002. **6**(5): p. 429-449.

[23] Visa, S. and A. Ralescu. Issues in Mining Imbalanced Data Sets - A Review Paper. in Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, MAICS-2005. 2005. Dayton.

[24] Murphey, Y.L., et al., OAHO: an Effective Algorithm for Multi-Class Learning from Imbalanced Data, in Proceedings of International Joint Conference on Neural Networks,. 2007: Orlando, Florida, USA.

[25] Guo, X., et al. On the Class Imbalance Problem. in Fourth International Conference on Natural Computation, 2008. ICNC '08. 2008.

[26] Dietterich, T.G., Ensemble Learning, in The Handbook of Brain Theory and Neural Networks, Second Edition, M.A. Arbib, Editor. 2002, The MIT Press: Cambridge.

27] Optiz, D. and R. Maclin, Popular Ensemble Methods: An Empirical Study. Journal of Articial Intelligence Research, 1999(11): p. 169-198.

[28] Kuncheva, L.I., Combining classifiers: Soft computing solutions, in Pattern Recognition: From Classical to Modern Approaches, World Scientific, S.K. Pal and A. Pal, Editors. 2001. p. 427-451.

[29] Polikar, R. (2009) Ensemble Learning. **4**.

[30] King, R.D., et al., Is it better to combine predictions? Protein Engineering, 2000. **13**(1): p. 15-19.

[31] Waegeman, W., et al., Supervised learning algorithms for multi-class classification problems with partial class memberships. Fuzzy Sets and Systems, 2011. **184**(1): p. 106-125.

[32] Takahashi, K., H. Takamura, and M. Okumura, Direct estimation of class membership probabilities for multiclass classification using multiple scores. Knowl. Inf. Syst., 2009. **19**(2): p. 185-210.

[33] Etzold, D., Improving spam filtering by combining Naive Bayes with simple k-nearest neighbor searches. The Computing Research Repository, 2003. **312004**.

[34. Kotsiantis, S. and P. Pintelas, Mixture of Expert Agents for Handling Imbalanced Data Sets. Annals of Mathematics, Computing & TeleInformatics, 2003. **1**(1): p. 46-55.

[35] Kotsiantis, S., K. Patriarcheas, and M. Xenos, A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. Knowledge-Based Systems, 2010. **23**(6): p. 529-535.

[36] Sainin, M.S. and R. Alfred, Nearest Neighbour Distance Matrix Classification, in International Conference on Advanced Data Mining and Applications (ADMA2010). 2010: Chongqing, China.

37] Asuncion, A. and D.J. Newman, UCI Machine Learning Repository [*http://www.ics.uci.edu/~mlearn/MLRepository.html]*. 2007, University of California, School of Information and Computer Science. : Irvine, CA.

[38. Alcalá-Fdez, J., et al., KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. . Journal of Multiple-Valued Logic and Soft Computing 2011. **17**(2-3): p. 255-287.

[39] Keogh, E., Xi, X., Wei, L. and Ratanamahatana, C. A., The UCR Time Series Classification/Clustering Homepage: *www.cs.ucr.edu/~eamonn/time_series_data/*. 2006.

[40] NIPS. NIPS Feature Selection Challenge. 2003 [cited 2011 2/2/2011]; Available from: http://www.nipsfsc.ecs.soton.ac.uk/datasets/.

[41] Weinberger, K.Q. and L.K. Saul, Distance Metric Learning for Large Margin Nearest Neighbor Classification. Journal of Machine Learning Research, 2009. 10: p. 207-244.

TABLE V. BENCHMARK DATASET DESCRIPTION.

Data	Reference	#Examples	#Att	#Class	#Min	#Max	Ratio	Domain
Wine	UCI/KEEL	178	13	3	48	71	1:1.48	Physical
Hayes	UCI/KEEL	132	4	3	30	51	1:2	Social
Contraceptive	UCI/KEEL	1473	9	3	333	629	1:2	Life
Balance	UCI/KEEL	625	4	3	49	288	1:6	Social
Dermatology	UCI/KEEL	366	34	6	20	112	1:6	Life
Statlog(Landsat)	UCI	5865	36	6	56	1072	1:19	Physical
Glass	UCI	209	9	7	4	76	1:19	Physical
Car	UCI	1728	6	4	65	1210	1:19	Business
Thyroid	UCI	7200	21	3	351	6666	1:19	Life
New Thyroid	UCI	193	5	3	8	150	1:19	Life
Nursery	UCI	12857	8	4	227	4320	1:19	Social
Lympography	UCI	148	18	4	2	81	1:41	Life
Ecoli	UCI/KEEL	336	7	8	2	143	1:72	Life
Yeast	UCI	1484	8	10	5	463	1:93	Life
PageBlocks	UCI/KEEL	5473	10	5	28	4913	1:175	Computer
Statlog(Shuttle)	UCI	58000	9	7	10	45586	1:4559	Physical