Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Meta-learning for imbalanced data and classification ensemble in binary classification

Sung-Chiang Lin [a,*], Yuan-chin I. Chang [b], Wei-Ning Yang [a]

[a] *Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan*
[b] *Institute of Statistical Science, Academia Sinica, Taipei, Taiwan*

ARTICLE INFO

Article history:
Received 17 November 2008
Received in revised form
10 April 2009
Accepted 25 June 2009
Communicated by Y. Fu
Available online 18 August 2009

Keywords:
Area under ROC curve
Fisher (linear) discriminant analysis
Imbalanced data
Meta-learning
ROC curve

ABSTRACT

To conduct binary classification with highly imbalanced data is a very common problem, especially when the examples of interest are relatively rare. In this paper, we proposed the "Meta Imbalanced Classification Ensemble (MICE)" algorithm in order to dilute the effect of imbalanced data. In the MICE, the majority group is partitioned based on the transformed features from "inner product" to retain the geometric relation between two groups. The empirical results show that the performance of MICE is better than some renowned classification methods in terms of the specificity and the sensitivity.

1. Introduction

The classification problems based on the imbalanced training data set occur often in applications when the events of interest are rare. That is, the size of the interesting minority group is usually in a rather small proportion in the training data set [9,32]. Many techniques are proposed to solve this kind of a binary classification problem through either data [14] or algorithmic levels (see [9,18]). The data level approach is usually based on re-sampling methods, including over-sampling the minority group (the group of interesting rare examples), or under-sampling the majority group (the group with large example sizes) [8,14,20]. The algorithmic level introduces unequal weights for two groups in the training strategy to force the classifier to pay more attention on the minority group [4,17,21]. However, the success of the re-sampling method may rely on how well the training samples represent the true population, while the success of the unequal weights approach may rely on parameter tuning processes.

In this paper, we proposed a meta-learning procedure that relies on partitioning the majority group and the integration of the meta-information of the sub-classifiers trained with the partitions of the majority one versus the minority one. This procedure can be viewed as an algorithmic level technique, where we eliminate the effect of class-imbalance by the techniques of decomposing and integrating. To prevent the complicated parameter tuning, the sub-classifiers in MICE can be just simple linear classifiers, such as Fisher's linear discriminant analysis (LDA) or the SVM with a linear kernel [2]. (Note that these two linear classifiers basically require no tuning parameters.) We found that the performance of MICE is very competitive and can even outperform many popular nonlinear classifiers such as the weight-adjusted SVM with a Gaussian kernel or the boundary adjusting to prevent its skewness toward the minority.

The partition with transformed features and logistic regression are two important steps in MICE. The first key feature here is that the partition of the majority group is based on the transformed features resulting from the inner product of original features, thus we not only control the size of each sub-group close to that of the minority group, but also retain the geometric relation between the majority group and the minority group. This geometric relation cannot usually be retained by a simple re-sampling scheme or a naive partitioning/clustering algorithm.

The second key feature of MICE is that using a logistic regression constructs the final ensemble. As the scales of function values of sub-classifiers are all different, the logistic regression can transform these function values of sub-classifiers into probabilities. Moreover, to integrate these sub-classifiers with a logistic regression, we can even apply statistical model selection techniques, such that the final classifier can be simplified further, and the testing time is shortened.

* Corresponding author.
E-mail addresses: lschiang@gmail.com (S.-C. Lin), ycchang@sinica.edu.tw (Y.-c. I. Chang), yang@cs.ntust.edu.tw (W.-N. Yang).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.06.015

In the following section, we state the proposed algorithm first, and discuss the details of each component step. After that, empirical results based on the synthesized and some benchmark data sets are reported and a brief conclusion remark follows.

2. Meta-learner

Meta learning is defined as a learning algorithm that is applied to the "learned meta-knowledge" [1,24], and a technique for assembling many (sub-)classifiers to improve the performance of component classifiers [10,12,31]. In other words, the meta-learning approach is to compute a number of classifiers first, and to integrate these classifiers in some fashion in order to boost overall performance [23]. Thus, there are at least two steps in a meta-learning algorithm: (1) sub-classifiers construction and (2) process of ensemble.

The proposed "Meta Imbalanced Classification Ensemble" (MICE) contains an extra pre-process step to deal with the obstacle of the imbalanced data, which contains three steps:

Three steps of MICE:

A. Project and decompose majority group and retain the geometric relation between the majority and the minority groups.
B. Construct sub-classifiers and re-scale their corresponding function values.
C. Construct the final ensemble with suitable model selection method.

Fig. 1 illustrates the basic idea of MICE and the details of it are stated as Algorithm 1.

Algorithm 1. MICE:Meta Imbalanced Classification Ensemble

```
Input:
Majority group $G_1$: $X^p_{n_p \times d} = \{x^p_j\}_{j=1}^{n_p}$, $x^p_j \in R^d$
Minority group $G_2$: $X^n_{n_n \times d} = \{x^n_i\}_{i=1}^{n_n}$, $x^n_i \in R^d$
Training label: Y
Set: Initialize k = $\sharp G_1/\sharp G_2$ rounded to integer and $N = n_p + n_n$.
if $k \geq 2$ then
  $C_0 = X^n_{n_n \times d}$
  Transformed matrix of majority group:
  $Z_{n_p \times n_n} = X^p_{n_p \times d} \cdot (X^n_{n_n \times d})^T$
  Function:
  $C_1, \ldots, C_k$ = two-stage-clustering($Z_{n_p \times n_n}$)
  for $i = 1$ to k do
    Train a linear classifier $H_i$: $C_0$ and $C_i$
    Get function values $f_i$ of all samples, and fit a simple logistic
  regression using $\{f_i(x_j), Y_j\}_{j=1}^N$.
    for $j = 1$ to N do
      Get $p_i(x_j) = (1 + exp(-\beta_{i,0} - \beta_{i,1} f_i(x_j)))^{-1}$
    end for
  end for
else
  Apply Regular Binary Classification Method; Stop
end if
Fit a logistic regression (sigmoid function) with responses
  $(Y_i)_{i=1}^N$ and covariates $\{p_{ij}, j = 1, \ldots, k\}_{i=1}^N$.
Apply model selection, then use the selected model compute
  the final prediction $\{P_{Final}(x_i)\}_{i=1}^N$.
Using training set to determine the threshold T based on the
  Bayesian cutting point.*
if $P_{Final} > T$
  Classify examples to minority group
else
  Classify examples to majority group
end if
```

* (see [2, p. 208].)

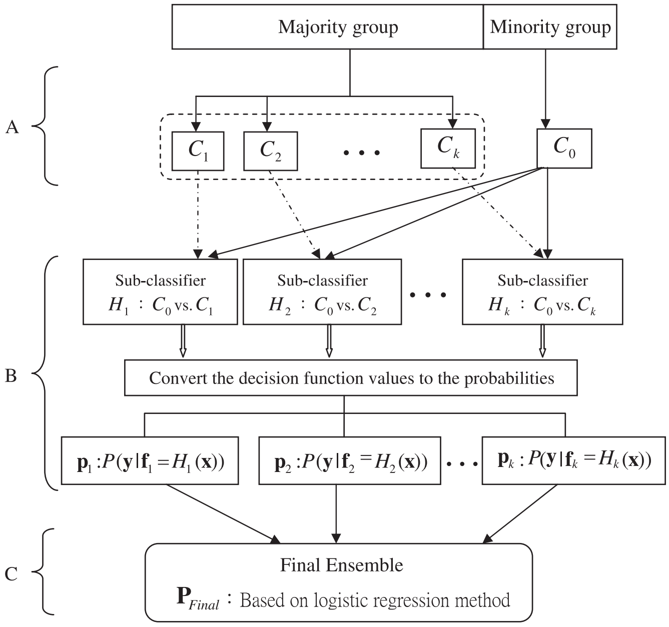
Majority group
Minority group
A
C_1
C_2
...
C_k
C_0
B
Sub-classifier H_1 : C_0 vs. C_1
Sub-classifier H_2 : C_0 vs. C_2
...
Sub-classifier H_k : C_0 vs. C_k
Convert the decision function values to the probabilities
$\mathbf{p}_1 : P(\mathbf{y}|\mathbf{f}_1 = H_1(\mathbf{x}))$
$\mathbf{p}_2 : P(\mathbf{y}|\mathbf{f}_2 = H_2(\mathbf{x}))$
...
$\mathbf{p}_k : P(\mathbf{y}|\mathbf{f}_k = H_k(\mathbf{x}))$
C
Final Ensemble
$\mathbf{P}_{Final}$: Based on logistic regression method

Fig. 1. Procedure for constructing Meta Imbalanced Classification Ensemble.

2.1. *Decomposition of majority group*

Numerous proposed methods for dealing with class-imbalanced problems are based on partitioning idea. To partition the majority group, when the training data are imbalanced, has been proposed by other authors [5,7,19]. However, they did not use the transformed features; which makes their results rather unstable in some cases due to the fact that the geometric relation between the majority and the minority groups cannot be retained. Here, we partition the majority group based on the "inner-product" transformed data which is novel and never found in literatures as far as we know.

In MICE, after partitioning based on the transformed features, the size of each sub-group of majority group is close to that of the minority group, while retaining the original geometric relation between the majority and the minority groups. In particular, we apply the clustering algorithm to the transformed features, which are "extracted" by conducting the inner product between the subjects of the majority group and the subjects of the minority group. This transformation can be viewed as projecting the points of the majority group on the space spanned by the points of the minority group. Because of nature of the inner-product, the transformed data contain the information of distances and angles between the subjects in the minority group and in the majority group. The angles between the majority group and the minority group can be viewed as the expression of related locations and then represent the related direction of the majority group to the minority group. This feature is the key of success of MICE. The partition procedure is as follows.

Remark 2.1. Only the data matrix of the majority group is transformed and the clustering algorithm is just applied to the transformed data matrix of the majority group. The minority data are only used in producing the inner product transformation of the majority group data points, but not involved in the clustering stage.

2.1.1. Partition under projection

Let x_j^p, $j=1,\ldots,n_p$ and x_i^n, $i=1,\ldots,n_n$ denote the feature vectors of the examples of the majority and the minority groups, respectively, where both x_j^p and x_i^n are d-dimensional vector in R^d, and n_p and n_n denote the sizes of these two groups. Let $z_{j,i}=\langle x_j^p, x_i^n\rangle$ for $i=1,\ldots,n_n$. Then, each subject in the majority group can be represented by a n_n-tuple vector $z_j=(z_{j,1},\ldots,z_{j,n_n})^T$, $j=1,\ldots,n_p$, where the notation w^T denotes the transpose of a vector w. That is, if $X^p_{n_p\times d}$ and $X^n_{n_n\times d}$ are the feature matrices of the majority and the minority groups. Then, the transformed feature matrix of the majority group can be written as $Z_{n_p\times n_n}=X^p_{n_p\times d}\cdot(X^n_{n_n\times d})^T$. Then, we apply a clustering algorithm to this new data matrix $Z_{n_p\times n_n}$.

According to the Cauchy–Schwartz inequality, the values of unimportant features will be diluted under this kind of an operation. Therefore, the clustering based on the transformed features will depend on the data of the minority group. This is exactly what we want here in order to retain the geometric relation between two groups as well as equalizing the ratio of the sizes of the sub-group of the major one to that of the minor one for constructing the local sub-classifiers. The numerical results show that the results based on the transformed data out-perform those based on the original data.

Fig. 2 shows the effect of the inner product transformation to the clustering. In this figure, we use three examples to illustrate

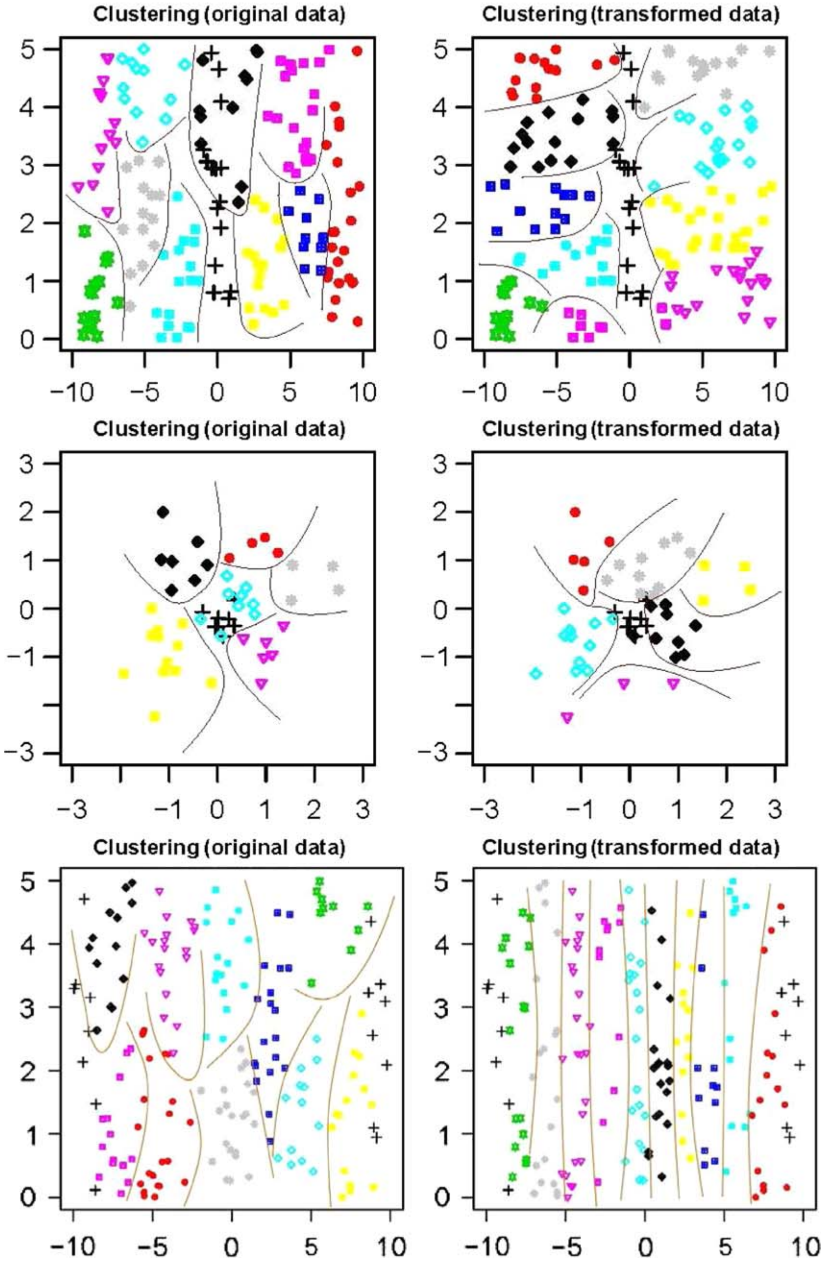

Fig. 2. The "+" sign denotes the data points of the minority group which are superimposed after clustering (but do not join the clustering process). All others denote the data points of the majority group where the different colors and shapes of points denote the sub-groups after clustering. Note that points with the same color but different shape are clustered into different sub-groups. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the effect. The clustering results based on the original features are in the left column and the results based on their inner-product transformed counterparts are on their right hand side.

In the first row, the original data are generated uniformly in a rectangle. Only the points marked by "+" denote the minority group and all others belong to the majority group. We use both colors and shapes of points to denote the clusters obtained by the *k*-means algorithm (i.e. points with the same color but different shape are treated as in different clusters). Note that the number of clusters—*k* is determined by the ratio of the sizes of two groups to make the sizes of two groups tend to equalize. The clusters in the left plot of the first row are in the stripe shape and somewhat "parallel" to the minority group, while on the right hand side, the clusters of majority groups are scattered in a "star" shape centering on the minority group.

A similar situation also applies to the plots in the second row. Here, the data are generated from concentric circles, and the minority group is right in the center. The clusters obtained from the transformed features are sill circling around the minority group, while the clusters obtained from the original features are not. The third row shows the situation in which the minority group is multimodal. These three examples explain how the clusters after transformation can help us to locate the "classification" boundary of the minority group, which cannot be offered by just any naive clustering using the original features.

If the original features are used directly, then some clusters may cross the minority group, as seen in the plots in the left column of Fig. 2. That is, in these cases, the clusters based on the original features mix up the information from the minority group, and consequently, and are less useful for constructing an ensemble. This situation will jeopardize the performance of the final ensemble.

In our empirical study, we use the *k*-means algorithm, a kind of the partitional/non-hierarchical clustering algorithms, to serve the clustering purpose. However, the *k*-means algorithm is a local searching procedure and the final sub-groups highly depend on the initial values [2]. In contrast to the partitional/non-hierarchical clustering algorithms, the drawback of hierarchical clustering is that the sample is never reallocated, when it is assigned to a cluster. Therefore, to make it more robust, a two-stage method (see [25,27,29]), which is a combination of hierarchical and non-hierarchical clustering methods, is used to find the final desired set of sub-groups. In the first stage, we apply a bottom-up hierarchical clustering method to look for *k* groups and their centroid, which are initial values for the next stage, where *k* is the number of groups decided by the sample sizes ratio of the majority to the minority groups. Using the means of these *k* sub-groups as the initial values, we then apply the *k*-means clustering algorithm to improve the stability of *k*-means.

2.2. Probability versus function value

For combining different classifiers, the scales of the decision values is an important issue. This is another key feature of MICE algorithm which takes full advantage of the logistic regression.

After partitioning the majority group, the sub-classifiers are constructed by using the minority group versus each partition obtained. Thus, there are *k* sub-classifiers, and each subject now can be represented as a *k*-dimensional vector based on their corresponding function values of these sub-classifiers (see also Section 2.3). That is, suppose that $\{f_i\}_{i=1}^k$ are *k* decision functions of the sub-classifiers $\{C_i\}_{i=1}^k$, respectively. Then, we can represent x_j by its function values $(f_1(x_j), \ldots, f_k(x_j))$. By fitting a simple logistic model, this vector of function values can be converted into a vector of probabilities.

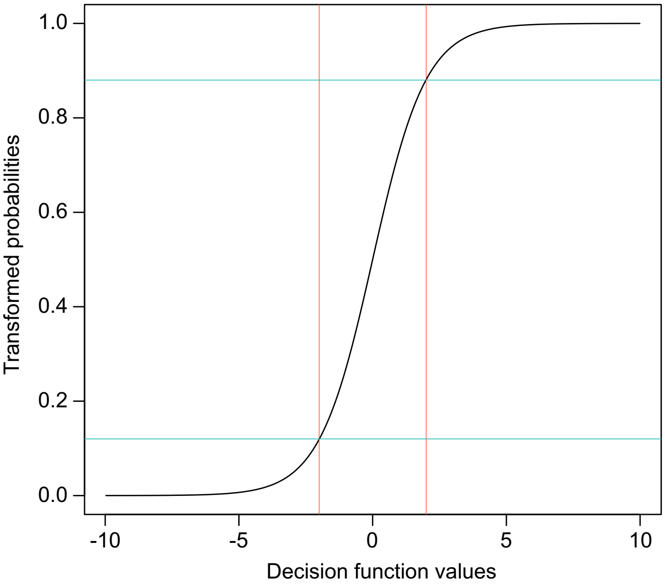

Fig. 3. An example of the probability transformation of the decision function values; the margin is enlarged under the logistic transformation. Note that the curve is convex first then becomes concave.

The scales of function values of meta-learners are all different, and the logistic transformation can help us to transform these function values of meta-learners into probabilities which not only serves as the re-scaling mechanics but also enlarges the "margin" of sub-classifiers in most cases.

Fig. 3 shows that the width between the two red lines denotes the "margin" of the decision function and the width between the two blue lines is the "margin" after logistic transformation. It can be seen from this plot that the "margin" is enlarged. However, this is not always the case for just any probability transform function. Intuitively, the logistic function works, because it is convex before its inflection point and concave after that. (It can be shown in terms of Mahalanobis distance, when FDA is used as the sub-classifier.) For further discussion about probability transformation of decision function values, refer to [13,30].

Note that when the decision function values of two (sub)-groups are not overlapped (i.e. completely separate), the estimates of the logistic regression coefficients are not well-defined. In this case, we suggest the reader to either re-define the estimates under this situation [33] or apply the Bayesian logistic regression with a non-informative prior [15].

In our study, Fisher's LDA and SVM with linear kernel are used for constructing sub-classifiers. Of course, the LDA and SVM-linear can be replaced by other linear classifiers or even some nonlinear classifiers. The reason why we use linear classifiers is not only to prevent the complex parameter tuning, but also because the final ensemble will be constructed by a logistic (nonlinear) model such that the complicated nonlinear classification boundary can be achieved at the ensemble stage. Therefore, it is not needed to use complicated sub-classifiers.

2.3. Final ensemble

As mentioned before, by applying the *k* sub-classifiers to all data, each sample point x_j can be represented by its corresponding probabilities, $(p_{j1}, \ldots, p_{jk})$, then using the original label y_j as the responses and the probabilities $\{\mathbf{p_j'} = (p_{j1}, \ldots, p_{jk})\}$, $j \geq 1$, as explanatory variables fit a logistic regression (sigmoid) function.

The model selection technique is used in order to reduce the number of clusters in the final ensemble. All the redundant clusters, such as the cluster masked by other clusters, are dropped. For simplicity, here, only the *p*-value for each single variable is used as the selection criterion. Other complicated model selection criterions are also possible to be used with the proposed method. The cutting threshold for the final ensemble is based on a standard logistic regression model with Bayesian consideration of the sample sizes ratio of the majority group to the minority group (see [2]).

For testing a new example, we need to transform its original feature values to probabilities, then plug in its probability representation to the model of the final ensemble and compare its value to the threshold. Note that it is not needed to apply all the sub-classifiers to the new sample point. Only those included in the final ensemble are applied. Thus, the testing time can be shortened.

3. Numerical results

3.1. Measures of performance

The measures of classification performance are briefly described as follows. Here we use AUC, accuracy, specificity, sensitivity (recall) to access the performance of classifiers. There are many others proposed for different needs of applications, such as the geometric mean, F-measure, etc. [3,11]. The predictive accuracy is the most commonly used performance measurement but it is not enough in applications with class imbalance. Since it may be overwhelmed by the major group and ignore the minor one. Except for the AUC, the other performance measures can be summarized as follows:

Let the threshold of classifier be fixed, then its results can be summarized as a 2×2 table (or confusion matrix), see Table 1:

- Accuracy: $(a+d)/(a+b+c+d)$.
- Specificity (true negative rate): $a/(a+b)$.
- Sensitivity or recall (true positive rate): $d/(c+d)$.

Table 1
The 2×2 contingency table (confusion matrix).

True label	Predicted label	
	Negative	Positive
Negative	a	b
Positive	c	d

3.2. Compared approaches

In the numerical study, we apply MICE to some synthesized data as well as some benchmark data. Fisher's LDA and SVM-linear classifier are used as the base linear classifiers. The SVM-linear classifier is based on the default values in LIBSVM: a library for support vector machines (http://www.csie.ntu.edu.tw/~cjlin/libsvm), which is the package "e1071" in R language (http://cran.r-project.org/). In comparison, six baseline approaches were used, including non-weighted SVM with Gaussian kernel (SVM(Gaussian)), weighted SVM with Gaussian kernel (weight SVM(Gaussian)), naive logistic regression, Fisher's LDA without feature transformation (Fisher's LDA(no extracted)), over-sampling with AdaBoost and under-sampling with AdaBoost. Both the weighted SVM with Gaussian kernel and the non-weighted SVM with Gaussian kernel tune the parameters via the function "tune.svm" in R language with/without setting the "class.-weights" parameter for asymmetric class sizes in SVM function equal to the value of the inversely proportional to the class sizes. In addition, naive logistic regression means classifying samples based on general logistic regression without any pre-process of data/features or any assembled approaches. Similarly, Fisher's LDA(no extracted) stands for Fisher's LDA as a meta-learner without feature transformations. Additionally, over-sampling and under-sampling approaches, which were proposed to solve imbalanced problems in past years, are also compared to the MICE algorithm. We adopted the over-sampling method with the AdaBoost approach as the assembled learner and the under-sampling method with AdaBoost (EasyEnsemble algorithm) proposed by Liu et al. [20].

We found that the results of MICE are very competitive and can even outperform the SVM Gaussian kernel classifier with weight adjusted, the over-sampling approach, and the under-sampling approach for data imbalance in some cases.

3.3. Synthesized data set

Table 2 summarizes the results of eight different methods which are the MICE with LDA or SVM-linear classifier as based learner, SVM Gaussian kernel (with/without classes weights), the naive logistic regression, the MICE with Fisher's LDA(no extracted), over-sampling with AdaBoost and under-sampling with AdaBoost. The results are based on 1000 runs. Five types of the synthesized data are used and illustrated in Fig. 4, except for the data set (IV).

In the first data set (I), the first variable in the minority group is generated from the uniform distribution $U(-1, 0.25)$, and in the majority group is from $U(-0.25, 10)$. Second variables in both groups are from $U(0, 5)$. The region of this data set overlaps between the minority and the majority groups. It is clear that the data set (I) is the simplest case among these five cases, and all the classifiers perform well in terms of accuracy. However, the specificity and sensitivity of the SVM-Gaussian-kernel without taking the weights of group sizes into account fail. The specificity and sensitivity are important measures of diagnostics (classification) which are commonly used in medical or biological related sciences, especially, in classification of rare samples (rare diseases or rare events).

The first variable in the next data set (II) in Fig. 4 is generated from the $U(-1.5, 1.5)$ for the minority group and from $U(0.5, 10)$ and $U(-10, -0.5)$ for the majority group, and second variables are from $U(0, 5)$ for both groups. The results of this example are also shown in Tables 2 and 3. In data set (II), even though all methods demonstrate fair accuracies and specificities, both simple logistic regression and SVM-Gaussian-kernel methods fail to identify the minority group owing to low sensitivities. In addition, the score of AUC from simple logistic regression is around 0.5; that is as poor as tossing a coin.

Data set (III) is concentric circles generated from the standard normal distribution in which the region also overlaps between the minority and the majority groups. Data set (IV) is generated using the same model as in [16], which is a 10-dimension concentric sphere and cannot be shown in Fig. 4. From the information supplied, it is evident that most performance measures are greater than 0.8 in both data sets, except for simple logistic regression resulting in the score of AUC being approximately 0.5 and the sensitivity being near 0 in the data set (III), and four measures being around 0.5 in the data set (IV).

Table 2
Mean (standard deviation) for testing results of MICE without model selection scheme and a Bayesian threshold.

	AUC	Accuracy	Specificity	Sensitivity (recall)
Synthesized data set (I)				
MICE with Fisher's LDA	99.05 (0.24)	95.02 (0.9)	94.60 (1.01)	99.20 (1.49)
MICE with SVM(linear)	99.06 (0.24)	95.05 (0.92)	94.64 (1.02)	99.18 (1.58)
SVM(Gaussian)	96.93 (2.22)^{a,b}	96.16 (0.66)	98.15 (0.69)	76.26 (7.10)
Weight SVM(Gaussian)	98.78 (0.38)^{a,b}	93.10 (1.17)	92.42 (1.29)	99.87 (0.59)
Logistic regression	99.11 (0.23)^{a,b}	94.80 (0.93)	94.29 (1.03)	99.93 (0.43)
Fisher's LDA(no extracted)	98.66 (0.63)^{a}	88.98 (1.39)	87.89 (1.53)	99.96 (0.34)
Over-sampling	97.75 (0.66)^{a,b}	95.52 (1.04)	95.17 (0.99)	99.05 (5.57)
Under-sampling	98.80 (0.34)^{a,b}	95.14 (0.93)	94.68 (1.03)	99.18 (0.95)
Synthesized data set (II)				
MICE with Fisher's LDA	96.22 (0.62)	88.62 (1.51)	87.58 (1.69)	99.07 (1.76)
MICE with SVM(linear)	96.23 (0.64)	88.67 (1.47)	87.63 (1.65)	99.10 (1.70)
SVM(Gaussian)	95.25 (1.12)^{a,b}	92.12 (0.95)	96.45 (1.21)	48.85 (12.77)
Weight SVM(Gaussian)	96.02 (0.72)^{a,b}	87.24 (1.62)	86.10 (1.86)	98.66 (2.22)
Logistic regression	50.05 (3.98)^{a,b}	79.33 (10.65)	85.91 (13.04)	13.47 (14.21)
Fisher's LDA(no extracted)	94.12 (2.05)^{a}	80.04 (4.60)	78.12 (5.08)	99.26 (1.82)
Over-sampling	94.48 (1.02)^{a,b}	90.43 (1.24)	89.61 (1.37)	98.55 (2.30)
Under-sampling	95.86 (0.72)^{a,b}	89.67 (1.29)	88.68 (1.44)	99.03 (1.43)
Synthesized data set (III)				
MICE with Fisher's LDA	98.72 (0.25)	94.13 (0.75)	95.00 (0.80)	90.58 (2.59)
MICE with SVM(linear)	98.70 (0.26)	94.12 (0.76)	95.01 (0.79)	90.53 (2.60)
SVM(Gaussian)	98.27 (0.63)^{a,b}	94.20 (0.79)	96.67 (0.74)	84.23 (2.87)
Weight SVM(Gaussian)	98.35 (0.61)^{a,b}	93.26 (0.81)	92.21 (1.06)	97.46 (1.56)
Logistic regression	50.06 (1.95)^{a,b}	79.82 (1.40)	99.67 (0.68)	0.00 (0.00)
Fisher's LDA(no extracted)	98.28 (0.58)^{a}	93.88 (0.83)	94.76 (0.92)	90.40 (2.82)
Over-sampling	95.19 (1.18)^{a,b}	91.95 (1.29)	91.51 (1.54)	93.68 (4.03)
Under-sampling	98.22 (0.44)^{a,b}	92.16 (0.93)	95.12 (2.35)	91.41(1.29)
Synthesized data set (IV)				
MICE with Fisher's LDA	93.31 (1.76)	80.86 (2.78)	80.12 (3.04)	92.12 (3.39)
MICE with SVM(linear)	92.71 (1.82)	82.31 (3.26)	81.85 (3.57)	88.98 (3.93)
SVM(Gaussian)	99.15 (0.24)^{a,b}	97.44 (0.63)	99.59 (0.20)	63.44 (8.07)
Weight SVM(Gaussian)	99.25 (0.25)^{a,b}	97.71 (0.57)	99.16 (0.40)	75.31 (9.05)
Logistic regression	49.95 (2.05)^{a,b}	50.21 (5.98)	50.54 (7.04)	49.14 (12.54)
Fisher's LDA(no extracted)	93.67 (1.97)^{a}	86.77 (2.67)	86.91 (2.80)	84.99 (4.53)
Over-sampling	85.07 (3.79)^{a,b}	89.71 (2.33)	91.65 (2.22)	61.81 (8.39)
Under-sampling	86.80 (4.67)^{a,b}	73.77 (8.59)	73.18 (8.89)	83.23 (6.52)
Synthesized data set (V)				
MICE with Fisher's LDA	99.08 (0.23)	93.78 (1.17)	93.20 (1.32)	99.50 (1.18)
MICE with SVM(linear)	99.05 (1.57)	93.16 (2.93)	92.50 (3.23)	99.82 (0.71)
SVM(Gaussian)	95.94 (2.18)^{a,b}	96.11 (0.71)	98.30 (0.70)	74.21 (6.58)
Weight SVM(Gaussian)	98.73 (0.42)^{a,b}	92.97 (1.18)	92.48 (1.33)	97.88 (2.54)
Logistic Regression	49.89 (3.50)^{a,b}	80.48 (10.50)	86.94 (12.67)	15.87 (13.10)
Fisher's LDA(no extracted)	96.02 (2.36)^{a}	81.62 (2.02)	80.06 (2.11)	97.22 (4.47)
Over-sampling	96.95 (0.85)^{a,b}	94.78 (0.94)	94.45 (0.93)	98.18 (4.97)
Under-sampling	98.81 (3.41)^{a,b}	94.39 (1.03)	93.92 (1.16)	99.14 (1.83)

[a] Stands for $p\text{-}value < 0.001$ which MICE with Fisher's LDA compare with the baseline approaches.
[b] Stands for $p\text{-}value < 0.001$ which MICE with SVM(linear) compare with the baseline approaches.

In the last synthesized data set (V), the first variable in the minority group is generated from $U(-10,-8.5)$ and $U(8.5,10)$, and in the majority group is from $U(-9,9)$. The second variables in both groups are also generated from $U(0,5)$. It is clear that the minority group is multimodal and the region also significantly overlaps between the minority and majority groups. As previous data sets, merely simple logistic regression performed worse than other methods in all measures.

Furthermore, we compare our method with the baseline methods by applying the paired t-test with significant level 0.001 to show the significant of the experiments. We only performed hypothesis testing for AUC criterion; hence, AUC is not affected by different thresholds. The testing results are demonstrated in Table 2. It is interesting that the MICE methods are statistically significantly better than the baseline methods in data (I), (II), (III) and (V). In addition, the performance with feature transformation is also significantly better than without feature transformation. Similarly, the performance of MICE is superior to over-sampling and under-sampling methods. Even though the scores of AUC of MICE methods are not as large as those of the baseline methods in the data set (IV), the sensitivities of MICE methods performed more excellent than theirs.

Although, the other three baseline methods, namely SVM(Gaussian), weight SVM(Gaussian) and logistic regression, still maintain high values of area under the ROC curve, they still cannot achieve the high specificity and the sensitivity, simultaneously. This indicates that even the weights of the sample sizes are taken into consideration, the SVM-Gaussian-kernel classifier still fails to choose a good threshold. Compared with the performance of the SVM-Gaussian-kernel classifier with weights, the performance of the LDA-based meta-learner is very competitive in all cases. The results of the SVM-linear-kernel-based meta-learner show that the MICE can accommodate different kinds of base classifiers. In this SVM-linear-kernel-based

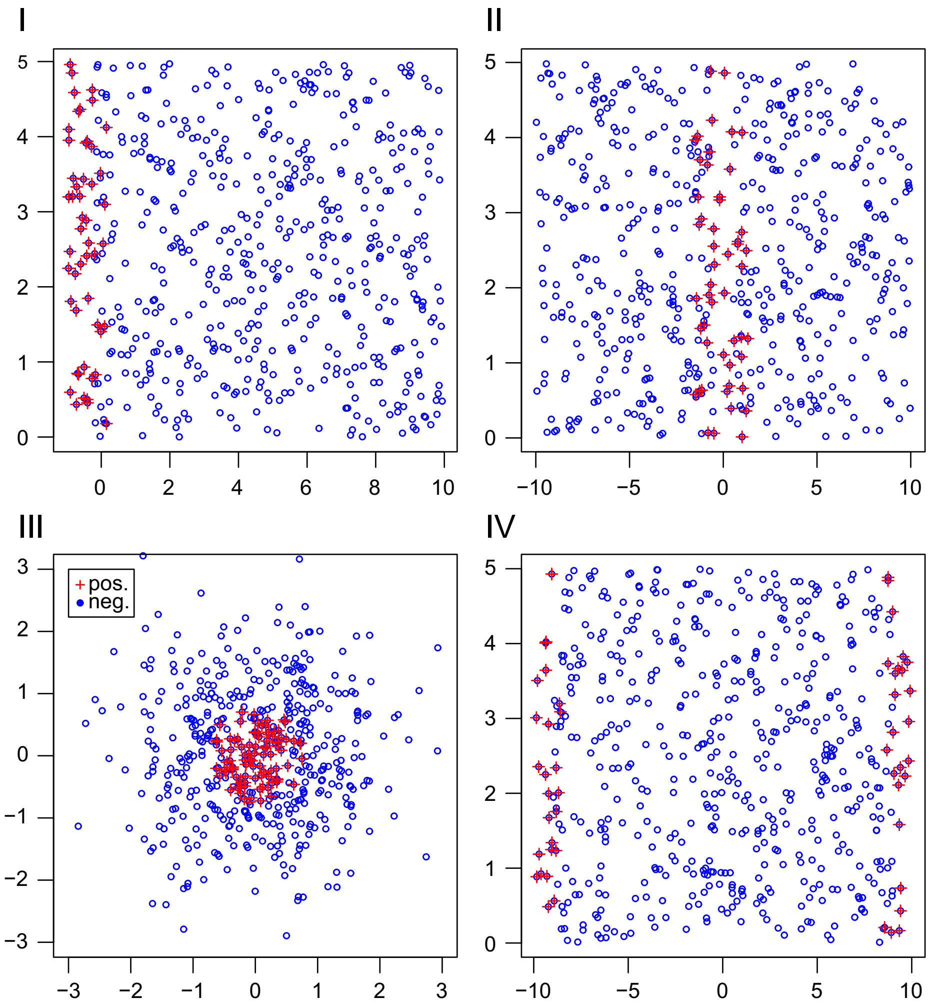

Fig. 4. Data plots of the synthesized data sets. In each graph, red dots denote the data points of the minority group and the blue dots are the data points of the majority group. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Mean (standard deviation) for testing results of MICE with model selection scheme and a Bayesian threshold.

Data	AUC	Accuracy	Specificity	Sensitivity
Synthesized data set (I)				
MICE with Fisher's LDA	98.96 (0.28)	95.08 (0.96)	94.80 (1.09)	97.89 (2.78)
MICE with SVM(linear)	98.80 (1.02)	94.77 (2.22)	94.49 (2.37)	97.55 (3.35)
Synthesized data set (II)				
MICE with Fisher's LDA	96.06 (0.80)	89.95 (1.95)	89.53 (2.28)	94.24 (5.17)
MICE with SVM(linear)	93.89 (9.31)	87.84 (9.36)	87.31 (10.24)	93.19 (6.43)
Synthesized data set (III)				
MICE with Fisher's LDA	98.72 (0.23)	94.15 (0.71)	95.04 (0.76)	90.57 (2.52)
MICE with SVM(linear)	98.69 (0.26)	94.13 (0.71)	95.03 (0.74)	90.49 (2.51)
Synthesized data set (IV)				
MICE with Fisher's LDA	92.22 (1.63)	79.91 (2.55)	79.17 (2.82)	90.36 (3.43)
MICE with SVM(linear)	91.73 (1.83)	81.38 (3.23)	80.92 (3.58)	87.26 (4.29)
Synthesized data set (V)				
MICE with Fisher's LDA	99.09 (0.24)	93.89 (1.08)	93.34 (1.22)	99.46 (1.23)
MICE with SVM(linear)	98.29 (6.09)	92.99 (2.8)	92.43 (2.69)	98.6 (9.54)

meta-learner, no parameter tuning procedure is required and can still perform as well as the weight-adjusted SVM-Gaussian-kernel classifier does.

The other method, namely Fisher's LDA(no extracted), in Table 2 shows the result based on Fisher's LDA as a meta-learner without feature transformation. Comparing the MICE with Fisher's LDA based on feature transformation, the performance with feature transformation is better at the significant level of 0.001.

Additionally, even though two data level approaches, over-sampling and under-sampling, maintain brilliant performance on AUCs, sensitivities and specificities for the most part, the performance is not expectable when facing more complex or high

Table 4
Mean (standard deviation) of the number of the partitions and the selected sub-classifiers.

Data	Partition	Number of clusters selected	
		LDA	SVM linear
(I)	10.00 (0.00)	2.20 (1.30)	4.74 (3.98)
(II)	10.00 (0.00)	3.25 (1.27)	2.97 (1.30)
(III)	4.50 (0.50)	4.50 (0.50)	4.50 (0.50)
(IV)	16.26 (5.49)	13.14 (1.97)	13.44 (2.20)
(V)	10.00 (0.00)	9.99 (0.25)	9.78 (1.32)

Table 5
The upper portion is the number of features, sample size and ratio of two groups of each experimental data set; The lower portion is the number of partitions used in each data set as well as the number of the clusters selected in the corresponding final ensemble.

The numbers of features, sample sizes and ratios			
Data set	Features	Sample size	Ratio
Allbp	29	2800	20.1
Breast	9	699	1.9
German	24	1000	2.3
Haber	3	306	2.8
Hepa	19	155	3.8
Hypo	25	3163	20.0
Sick	29	3772	15.3
S-euth	25	3163	9.8
Abalone(Ring 7)	8	4177	9.7
Letter(A)	17	20000	24.3
Satimage(class 4)	37	6435	9.3
The numbers of partitions and the selected clusters			
	Partition	Num. of part. selected	
		LDA	SVM Linear
Allbp	21	7.38 (1.67)	8.54 (1.39)
Breast	2	2.00 (0.00)	1.89 (0.31)
German	3	2.01 (0.12)	2.54 (0.50)
Haber	3	2.00 (0.09)	1.95 (0.33)
Hepa	4	2.63 (0.62)	3.10 (0.70)
Hypo	21	8.10 (1.87)	7.94 (1.75)
Sick	16	5.19 (1.47)	6.59 (1.32)
S-euth	10	5.31 (0.92)	5.62 (0.95)
Abalone(Ring 7)	10	4.25 (0.51)	2.49 (0.56)
Letter(A)	25	13.83 (2.31)	14.34 (2.61)
Satimage(class 4)	10	7.39 (0.61)	6.59 (0.66)

Boldface means the data sets are highly imbalanced.

dimensional data sets such as data (IV); that is, the specificity and the sensitivity cannot achieve superiority simultaneously.

Model selection: By employing some model selection techniques with the combination of sub-classifiers becoming the final ensemble with a logistic regression, we can reduce the number of clusters used in the final ensemble, which makes the testing procedure more efficient. For simplicity, we use $p-values$ which are calculated via the Wald test by the package "brlr" in R language as the criteria of model selection in this paper; the significance level was 0.05. Table 3 summarizes the results of two meta-learners with model selection. Compared with Table 2, those only show a slight decline in some cases.

Table 4 states the average partitions and its standard deviation at each data set, and partitions selected by LDA-based and SVM-linear-based meta-learners with model selection. From this table, we found that if the geometric relation between two groups is rather complicated, then MICE with model selection tends to use more partitions. If the relation between two groups are relatively simple, then it will just use a few of them, which will make the testing more efficient. It can be seen that due to the dimensionality of the data set (IV), the variation of the number of partitions is much larger than those of others.

Here, we demonstrate that the complicated parameter tuning and the threshold choosing procedures can be omitted by using the proposed meta-learning procedure without sacrificing performance. Thus, this paper provides an alternative choice to advance classification algorithms, such as SVM, which may require a tuning procedure.

3.4. Some benchmark data

We apply MICE to some benchmark data sets: Allbp-op (Allbp), Wisconsin breast cancer (Breast), German, Haberman (Haber), Hepatitis (Hepa), Hypothyroid (Hypo), Sick, and Sick-euthyroid (S-euth). Except for binary data sets, we also adopt some multi-class data sets, abalone, letter and satimage, and re-define the classes into two classes; in other words, "Ring 7" in abalone is viewed as a minority group and other classes are categorized as a majority group; "Letter A" is viewed as a minority group and other letters are categorized as a majority group; "Class 4" in satimage is viewed as a minority group and other classes are categorized as a majority group. The first portion of Table 5 summarizes the number of features, sizes and the ratios of two groups of the data sets. The number of partitions produced by k-means for each data set, as well as the number of clusters, which are actually included in the final ensemble, is recorded in the second portion of Table 5. Due to the randomness of the simulation, the number of selected clusters may vary for each simulation run and their standard deviation is stated within parentheses. Again, it is based on a thousand runs for each data set. Each time, 90% of the data are randomly selected as a training data set, the stratified sampling scheme is used to maintain the group-ratio and their standard deviation is reported within the parentheses (see Table 5). Note that all the results of MICE here are with model selection.

Four performance measures are reported: area under ROC curve, accuracy, the specificity, and the sensitivity. Five methods are compared: the MICE with LDA or SVM-linear classifier as based learner, the SVM-Gaussian-kernel with weight adjusted for imbalanced data, over-sampling with AdaBoost and under-sampling with AdaBoost (EasyEnsemble). All experiments are done by using the packages in R, in which the SVM package is based on the original program of [6,26]. Table 6 summarizes the results of real data sets and we also compare the scores of AUC by applying the paired t-test with a significant level of 0.001 to show the significance of the experiments, as AUC is not affected by different thresholds.

It can be seen in Table 6 that if the accuracy is the solo measure, then the SVM-Gaussian with weights adjusted for the imbalanced data outperforms the other methods. But if the specificity and the sensitivity are considered as an important character of the performance, then the proposed MICE's (with LDA and with SVM-linear kernel classifier) are much better than the weight-adjusted SVM classifier with Gaussian kernel. In addition, over-sampling and under-sampling approaches seem to perform as well as MICE in some cases, while MICE has a higher value of AUC or balanced values between sensitivity and specificity.

Note that among these benchmark data sets, the ratios of the majority group size to the minority group size in Allbp-op, Hypothyroid, Sick and Sick-euthyroid are more than 9.8. In these data sets, the SVM Gaussian kernel classifier cannot successfully

Table 6
Mean (standard deviation) for testing results of MICE with model selection scheme and a Bayesian threshold.

Data sets	AUC	Accuracy	Specificity	Sensitivity
Allbp				
MICE with Fisher's LDA	88.73 (4.97)	83.47 (5.72)	83.49 (6.01)	83.14 (6.44)
MICE with SVM(linear)	90.24 (4.55)	88.57 (3.54)	89.04 (3.72)	79.66 (9.04)
Weight SVM(Gaussian)	62.61 (10.56)[a,b]	95.02 (0.00)	100.00 (0.00)	0.00 (0.00)
Over-sampling	90.62 (3.45)[a]	90.07 (2.45)	90.01 (2.63)	91.24 (6.98)
Under-sampling	89.42 (2.91)[a,b]	87.18 (2.72)	87.16 (2.86)	85.56 (8.46)
Breast				
MICE with Fisher's LDA	99.41 (0.49)	97.38 (1.64)	96.85 (2.23)	98.36 (2.57)
MICE with SVM(linear)	99.12 (0.88)	97.61 (1.55)	96.77 (2.30)	99.16 (1.94)
Weight SVM(Gaussian)	98.73 (1.29)[a,b]	96.75 (1.82)	95.83 (2.59)	98.46 (2.73)
Over-sampling	95.81 (2.64)[a,b]	92.37 (7.21)	93.28 (4.03)	90.71 (18.36)
Under-sampling	98.13 (1.23)[a,b]	94.94 (2.35)	94.23 (3.06)	96.26 (3.82)
Ger				
MICE with Fisher's LDA	79.31 (3.99)	75.78 (3.62)	82.19 (4.41)	60.81 (7.97)
MICE with SVM(linear)	78.70 (3.96)	75.49 (3.62)	82.28 (4.48)	59.65 (7.98)
Weight SVM(Gaussian)	76.07(4.23)[a,b]	72.51 (2.81)	93.64 (2.71)	23.20 (7.36)
Over-sampling	72.60 (5.08)[a,b]	69.32 (4.68)	70.51 (6.36)	65.54 (7.87)
Under-sampling	77.29 (4.43)[a,b]	68.67 (4.42)	73.72 (6.99)	66.50 (5.85)
Haber				
MICE with Fisher's LDA	72.22 (7.77)	74.95 (5.81)	87.64 (6.40)	42.51 (14.43)
MICE with SVM(linear)	72.28 (8.00)	74.97 (5.79)	87.18 (6.44)	43.76 (14.18)
Weight SVM(Gaussian)	68.60 (10.65)[a,b]	66.76 (8.57)	74.23 (9.53)	47.66 (18.71)
Over-sampling	64.03 (9.26)[a,b]	64.17 (8.08)	68.82 (11.05)	52.30 (13.55)
Under-sampling	67.97 (9.46)[a,b]	65.97 (7.96)	69.09 (10.60)	57.98 (13.65)
Hepa				
MICE with Fisher's LDA	82.26 (11.38)	77.10 (9.96)	77.79 (11.88)	74.85 (20.32)
MICE with SVM(linear)	83.35 (10.45)	78.75 (8.26)	79.88 (10.05)	75.08 (18.33)
Weight SVM(Gaussian)	83.96 (10.05)	76.48 (0.26)	100.00 (0.00)	0.05 (1.12)
Over-sampling	77.37 (12.57)[a,b]	73.75 (10.53)	76.98 (12.43)	63.28 (24.59)
Under-sampling	81.21 (9.34)[b]	71.48 (9.84)	70.11 (12.34)	74.55 (20.53)
Hypo				
MICE with Fisher's LDA	94.46 (3.00)	88.11 (3.47)	88.05 (3.63)	89.16 (8.18)
MICE with SVM(linear)	97.23 (1.63)	92.45 (2.36)	92.56 (2.50)	90.37 (6.90)
Weight SVM(Gaussian)	94.62 (2.65)[b]	93.94 (1.41)	95.04 (1.48)	73.13 (11.85)
Over-sampling	94.93 (2.94)[a,b]	93.30 (1.69)	93.49 (1.82)	89.73 (6.82)
Under-sampling	95.84 (2.43)[a,b]	86.55 (2.73)	86.31 (2.95)	90.08 (6.48)
Sick				
MICE with Fisher's LDA	94.37 (2.41)	89.13 (2.71)	89.10 (2.92)	89.57 (5.30)
MICE with SVM(linear)	95.25 (2.42)	90.01 (2.96)	90.07 (3.08)	89.04 (6.33)
Weight SVM(Gaussian)	66.55 (8.79)[a,b]	93.67 (0.00)	100.00 (0.00)	0.00 (0.00)
Over-sampling	93.88 (2.15)[a,b]	95.52 (1.07)	95.80 (1.13)	90.37 (4.45)
Under-sampling	95.05 (2.03)[a]	94.48 (2.02)	94.91 (2.25)	88.12 (7.16)
S-euth				
MICE with Fisher's LDA	93.82 (1.93)	85.29 (2.15)	84.58 (2.37)	92.06 (4.27)
MICE with SVM(linear)	94.17 (1.88)	86.88 (1.81)	86.27 (2.06)	92.66 (3.80)
Weight SVM(Gaussian)	90.29 (3.63)[a,b]	90.41 (2.01)	92.71 (3.05)	68.42 (15.20)
Over-sampling	93.33 (2.50)[a,b]	94.23 (1.21)	94.61 (1.26)	90.63 (4.29)
Under-sampling	94.11 (2.37)	93.79 (1.85)	94.10 (1.90)	90.81 (6.45)

[a] Stands for *p-value* < 0.001 which MICE with Fisher's LDA compare with the baseline approaches.
[b] Stands for *p-value* < 0.001 which MICE with SVM(linear) compare with the baseline approaches.

locate the rare events, even if the imbalance of data sizes are adjusted by the weight parameter, and the performance only slightly improves in that data with low group-size ratios. On the other hand, the two meta-learners (LDA-based MICE and SVM-linear-kernel-based MICE) perform very well. Both of them have rather balanced specificity and sensitivity. In Table 6, it can be seen that MICE does perform much better than the baseline approach on the Allbp and Sick data sets that are highly imbalanced. The result of another imbalanced data set, Hypo, shows that MICE based on the meta-learner, SVM(linear), also performs better than the baseline approach. Even though the performance relied on AUC of LDA-based MICE, it seems not be statistically significantly better. The values of sensitivity and specificity are quite balanced and MICE based on LDA results in greater sensitivity than the baseline approach that is why the MICE algorithm is proposed to improve the accuracy rate of the minority group. These results indicate that using weight adjusted alone in SVM may not be able to solve the highly imbalanced classification problems. Table 7 presents the results of multi-class as binary. The MICE algorithm performs better than other methods among AUC, sensitivity and specificity. In addition, Table 5 shows the number of classifiers in MICE based on model selection. Even though the MICE algorithm uses the less number of classifiers in the final stage, it still has desirable performance. That is, in highly imbalanced data sets (Tables 6 and 7), MICE uses no more than half of classifiers, but is very competitive and can even outperform other approaches. Hence, MICE is more efficient in the testing process.

Table 7
Mean (standard deviation) for testing results of MICE with model selection scheme and a Bayesian threshold (multi-class).

	AUC	Accuracy	Specificity	Sensitivity
Abalone(Ring 7)				
MICE with Fisher's LDA	86.37 (1.85)	78.18 (2.20)	77.83 (2.56)	82.50 (5.84)
MICE with SVM(linear)	77.92 (7.78)	67.56 (8.09)	66.33 (8.67)	79.19 (7.92)
Weight SVM(Gaussian)	83.54 (2.66)[a,b]	75.44 (2.01)	74.63 (2.33)	83.03 (5.94)
Over-sampling	81.60 (2.41)[a,b]	72.45 (2.72)	70.95 (3.13)	86.61 (4.78)
Under-sampling	84.51 (2.22)[a,b]	73.31 (2.24)	72.10 (2.53)	83.77 (4.26)
Letter(A)				
MICE with Fisher's LDA	99.51 (0.16)	96.61 (0.45)	96.66 (0.48)	95.44 (2.02)
MICE with SVM(linear)	98.71 (6.60)	95.03 (12.80)	95.04 (12.88)	96.31 (2.06)
Weight SVM(Gaussian)	99.98 (0.01)[a,b]	99.93 (0.06)	99.99 (0.03)	98.84 (1.20)
Over-sampling	96.24 (1.57)[a,b]	95.47 (0.72)	95.59 (0.76)	92.34 (2.65)
Under-sampling	99.62 (0.15)[a,b]	97.26 (0.59)	97.34 (0.61)	95.27 (2.08)
Satimage(class 4)				
MICE with Fisher's LDA	92.33 (1.12)	85.83 (1.28)	85.81 (1.42)	86.10 (4.01)
MICE with SVM(linear)	92.65 (1.15)	85.23 (1.44)	85.11 (1.68)	86.37 (4.25)
Weight SVM(Gaussian)	96.79 (0.91)[a,b]	93.53 (1.21)	94.54 (1.60)	84.25 (5.99)
Over-sampling	88.61 (2.22)[a,b]	84.72 (1.52)	85.01 (1.68)	82.08 (3.89)
Under-sampling	93.84 (1.08)[a,b]	84.21 (1.49)	83.84 (1.66)	87.34 (3.45)

[a] Stands for *p-value* <0.001 which MICE with Fisher's LDA compare with the baseline approaches.
[b] Stands for *p-value* <0.001 which MICE with SVM(linear) compare with the baseline approaches.

Remark 3.1 (*Probability transformation*). To use probability distributions as the output of meta-learners may allow us to use them not only in the predictions, but also as the confidence levels of the meta-learners. Here we apply a logistic regression to convert the decision function values to the probabilities which are similar to [13,30]. This transformation not only re-scales the decision values of different classifiers to [0, 1], but also makes the margin of transformed probabilities of meta-learners "relatively-larger" than the margin of the original decision values of meta-learners. When the function values of two groups follow normal distributions with the same variance and different means. The advantage of logistic regression transformation can follow easily from [22, p. 138]. For non-normal case, the results can be obtained through "normalization" of the original function values.

4. Conclusion

The binary classification with highly imbalanced data is studied in this paper. A meta-learning algorithm based on the inner-product transformed features, which retains the geometric relations among two groups, is proposed. In addition, the logistic regression method is used not only to re-scale the decision values to probabilities but also to construct the final ensemble classifier. Since the LDA (or SVM with linear kernel) classifier is used as a based sub-classifier and a Bayesian threshold is used to decide the final cutting point, no parameter tuning procedure is required in the proposed method. The numerical results show that the proposed method is very promising in terms of accuracy and area under the ROC curve. Furthermore, if the detection of the rare events is emphasized, as in the highly imbalanced classification problem, the proposed method is very stable and even outperforms the benchmark SVM classifier in terms of the specificity and the sensitivity, which are important measures used in detection of rare events (such as disease diagnostics, network intrusion, etc.). One more advantage is that the proposed method relies on many successful techniques in machine learning, such as Fisher's LDA [28], support vector machine (SVM [4]), *k*-means clustering methods [27,29] and logistic regression [22], which are well studied, and have related software available in many popular computing packages. More details can be found in related references. This may actually benefit some practitioners in other research areas. In addition, the multi-group classification can be solved by many binary classifiers and is a natural extension of this method. Moreover, the base classifier here can be replaced by a more advanced or complicated classifier developed by users for their needs, as long as the decision function values can be available, but it is out of the scope of this study.

References

[1] S. Ali, K.A. Smith-Miles, A meta-learning approach to automatic kernel selection for support vector machines, Neurocomputing 70 (2006) 173–186.

[2] E. Alpaydin, Introduction to Machine Learning, MIT Press, USA, 2004.

[3] R. Barandela, J.S. Sanchez, V. Garcia, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition 36 (2003) 849–851.

[4] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (1998) 955–974.

[5] P.K. Chan, S.J. Stolfo, Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1998, pp. 164–168.

[6] C.C. Chang, C.J. Lin, LIBSVM: a library for Support Vector Machines 〈http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.ps.gz〉, 2001.

[7] Y.-c.I. Chang, S.C. Lin, Synergy of logistic regression and support vector machine in multi-class classification, in: Proceedings of the IDEAL 2004, Lecture Notes in Computer Science, vol. 3177, Springer, Berlin, Heidelberg, 2004, pp. 132–141.

[8] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[9] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, in: Proceedings of the SIGKDD Explorations Newsletter, vol. 6, ACM SIGKDD Explorations, 2004, pp. 1–6.

[10] G.C. Christophe, R. Vilalta, P. Brazdil, Introduction to the special issue on meta-learning, Machine Learning 54 (2004) 187–193.

[11] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, A. Geissbuhler, Learning from imbalanced data in surveillance of nosocomial infection, Artificial Intelligence in Medicine 37 (2006) 7–18.

[12] T.G. Dietterich, Ensemble methods in machine learning, in: J. Kittler, F. Reli (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, Heidelberg, 2000, pp. 1–15.

[13] S. Dzeroski, B. Zenko, Is combining classifiers with stacking better than selecting the best one?, Machine Learning 54 (2004) 255–273.

[14] A. Estabrooks, A multiple resampling method for learning from imbalanced data sets, Computational Intelligence 20 (2004) 18–36.

[15] D. Firth, Bias reduction of maximum likelihood estimates, Biometrika 80 (1993) 27–38.

[16] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Annals of Statistics 28 (2000) 337–407.

[17] T. Imam, K.M. Ting, J. Kamruzzaman, z-SVM: an SVM for Improved classification of imbalanced data, in: A. Sattar, B.H. Kang (Eds.), AI 2006: Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence, vol. 4304, Springer, Berlin, Heidelberg, 2006, pp. 264–273.

[18] N. Japkowicz, Learning from imbalanced data sets: a comparison of various strategies, in: Tech Rep. WS-00-05, AAAI Workshop on Learning from Imbalanced Data Sets, AAAI Press, Menlo Park, CA, 2000.

[19] C. Li, Classifying imbalanced data using a bagging ensemble variation (bev), in: Proceedings of the ACM-SE 45, ACM, New York, 2007, pp. 203–208.

[20] X.Y. Liu, J. Wu, Z.H. Zhou, Exploratory under-sampling for class-imbalance learning, in: Proceedings of the Sixth International Conference on Data Mining, IEEE Computer Society, Washington, 2006, pp. 965–969.

[21] X.Y. Liu, Z.H. Zhou, The influence of class imbalance on the cost-sensitive learning: an empirical study, in: Proceedings of the ICDM '06, IEEE The Computer Society, 2006.

[22] P. McCullagh, Generalized Linear Models, second ed., Chapman & Hall, New York, 1989.

[23] A.L. Prodromidis, P.K. Chan, S.J. Stolfo, Meta-learning in distributed data mining systems: issues and approaches, in: Advances of Distributed Data Mining, AAAI Press, 2002.

[24] R.B.C. Prudencio, T.B. Ludermir, Meta-learning approaches to selecting time series models, Neurocomputing 61 (2004) 121–137.

[25] G. Punj, D.W. Stewart, Cluster analysis in marketing research: review and suggestion for application, Journal of Marketing Research 20 (1983) 134–148.

[26] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2007.

[27] A.C. Rencher, Methods of Multivariate Analysis, second ed., Wiley, USA, 2002.

[28] S. Sharma, Applied Multivariate Techniques, Wiley, New York, 1996.

[29] N.H. Timm, Applied Multivariate Analysis, Springer, New York, 2002.

[30] K.M. Ting, I.H. Witten, Issues in stacked generalization, Journal of Artificial Intelligence Research 10 (1999) 271–289.

[31] R. Vilalta, Y. Drissi, A perspective view and survey of metalearning, Artificial Intelligence Review 18 (2002) 77–95.

[32] J.H. Zhao, X. Li, Z.Y. Dong, Online rare events detection, in: Z.H. Zhou, H. Li, Q. Yang (Eds.), Advances in Knowledge Discovery and Data Mining, Lecture Notes in Artificial Intelligence, vol. 4426, Springer, Berlin, Heidelberg, 2007, pp. 1114–1121.

[33] J. Zhu, T. Hastie, Kernel logistic regression and the import vector machine, Journal of Computational and Graphical Statistics (2001) 1081–1088.

Sung-Chiang Lin receives his MBA degree in National Taiwan University of Science and Technology, Taipei, Taiwan, and is now a PhD student of Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan.

Yuan-chin Ivan Chang receives his PhD degree in Statistics from University of Illinois, Urbana Champaign, USA, and is now an associate research fellow of Institute of Statistical Science, Academia Sinica and associate professor of Department of Statistics, National Cheng-Chi University, Taipei, Taiwan.

Wei-Ning Yang receives his PhD degree in Industrial and Systems Engineering from The Ohio State University, Columbus Ohio, USA, and is now an associate professor in the Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan.